



# Méthodes et modèles de conception et d'évaluation des interfaces homme-machine

Christophe Kolski

## ► To cite this version:

Christophe Kolski. Méthodes et modèles de conception et d'évaluation des interfaces homme-machine. Interface homme-machine [cs.HC]. Université de Valenciennes et du Hainaut-Cambrésis, 1995. tel-01300869

**HAL Id: tel-01300869**

**<https://hal.science/tel-01300869>**

Submitted on 11 Apr 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# **Mémoire**

présenté à  
l'université de Valenciennes et du Hainaut-Cambrésis

pour l'obtention de

## **l'Habilitation à Diriger des Recherches**

spécialité  
INFORMATIQUE

par

**Christophe KOLSKI**

Docteur de l'université de Valenciennes  
Ingénieur EUDIL

## **Méthodes et modèles de conception et d'évaluation des interfaces homme-machine**

soutenu le 4 janvier 1995 devant la commission d'examen :

Mme Marie-France	BARTHET	Rapporteur
Mme Jocelyne	NANARD	Rapporteur
Mr Jean-Claude	ANGUE	Examineur
Mr Arnaud	FREVILLE	Examineur
Mr Alain	HAURAT	Rapporteur
Mr Patrick	MILLOT	Directeur
Mr Jean-Claude	SPERANDIO	Examineur

## Remerciements

Le travail ayant fait l'objet de ce mémoire d'Habilitation à Diriger des Recherches a été mené au Laboratoire d'Automatique et de Mécanique Industrielles et Humaines (LAMIH), U.R.A. C.N.R.S. 1775, à l'université de Valenciennes, dans l'équipe "Informatique industrielle et communication homme-machine" dirigée par le professeur Patrick Millot.

Mes remerciements les plus chaleureux vont naturellement au professeur Patrick Millot, mon directeur de recherche, qui m'a suivi, guidé et conseillé avec enthousiasme et rigueur tout au long de mes recherches, depuis mon arrivée au laboratoire en septembre 1985. Je lui suis en outre très reconnaissant de m'avoir fourni les moyens de mener des recherches dans d'excellentes conditions humaines et matérielles.

Je suis très reconnaissant envers Madame Marie-France Barthet, Professeur à l'université de Toulouse I, Madame Jocelyne Nanard, Professeur à l'université de Montpellier II, et Monsieur Alain Haurat, Professeur à l'université de Savoie Annecy, de m'avoir fait l'honneur d'examiner ce travail et d'en être les rapporteurs.

Je remercie vivement Monsieur Jean-Claude Angué, Professeur à l'université de Valenciennes, Monsieur Arnaud Fréville, Professeur à l'université de Valenciennes, et Monsieur Jean-Claude Spérandio, Professeur à l'université de Paris V, pour leur participation à ce jury.

Je souhaite également remercier tous ceux avec qui j'ai eu le plaisir de travailler en étroite collaboration, dans le cadre d'encadrement de thèses ou de DEA. Je pense plus particulièrement à François Gambiez, Martial Grislin, Emmanuelle Le Strugeon, Faouzi Moussa, Thierry Poulain, Mustapha Tendjaoui et Bertrand Vilain.

De nombreuses autres personnes ont contribué, directement ou non, et chacun à leur manière, à l'aboutissement de ce mémoire. Je les en remercie. J'ai une pensée spéciale à ce sujet pour Mouldi Sagar et René Mandiau pour leurs remarques pertinentes.

*Je dédie ce mémoire à Jutta, Nouf-nouf et Panzi qui ont su faire preuve d'une patience infinie durant la rédaction de ce mémoire, à toute ma famille pour ses encouragements chaleureux, ainsi qu'à ma belle-famille pour ses modules d'assistance à la rédaction (Kuchen, gute Luft und Kaffee).*

*Remarque préalable : ce mémoire datant de 1995 a été reconstitué en 2016. Pour des raisons d'incompatibilité partielle avec le format des logiciels d'édition de texte et de figures initialement utilisés, il a été nécessaire de refaire certaines figures et d'adapter légèrement la mise en page.*

# Sommaire

<b><i>Première partie : Curriculum vitae.....</i></b>	<b>6</b>
---	----------

Renseignements personnels.....	7
Activités de recherche.....	8
Encadrement de thésards et DEA.....	9
Liste des travaux, ouvrages, articles et réalisations.....	10
Activités d'enseignement.....	14
Activités d'administration et autres responsabilités collectives.....	16

<b><i>Seconde partie : Méthodes et modèles de conception et d'évaluation des interfaces homme-machine, application aux systèmes industriels complexes.....</i></b>	<b>17</b>
--	-----------

<b><i>Introduction générale.....</i></b>	<b>18</b>
--	-----------

<b><i>Chapitre 1. Etat de l'art sur le domaine de recherche.....</i></b>	<b>17</b>
--	-----------

1.1. Positionnement de la recherche sur le plan méthodologique .....	24
1.1.1. Etude critique des principaux modèles et méthodes de développement issus du génie logiciel.....	24
1.1.2. Cadre théorique et méthodologique pour le développement des interfaces homme-machine.....	30
1.2. Analyse et modélisation des tâches humaines et de l'utilisateur .....	33
1.2.1. Analyse et modélisation des tâches humaines.....	33
1.2.2. Analyse et modélisation cognitive de l'utilisateur.....	38
1.2.3. Conclusion sur l'analyse et la modélisation des tâches et de l'utilisateur.....	45
1.3. Modèles d'architecture de l'interface homme-machine.....	46
1.3.1. Présentation de quatre modèles représentatifs.....	46
1.3.2. Positionnement de concepts d'interfaces "intelligentes" dans le modèle de SEEHEIM.....	50
1.3.3. Conclusion sur les architectures d'interface.....	55
1.4. Spécification, réalisation et évaluation des interfaces homme-machine.....	56
1.4.1. Spécification des interfaces homme-machine.....	56
1.4.2. Outils graphiques de réalisation des interfaces homme-machine.....	60
1.4.3. Méthodes utilisables pour l'évaluation des interfaces homme-machine.....	62

Conclusion.....	69
Bibliographie du chapitre 1.....	71

## **Chapitre 2. Contribution aux méthodes et modèles de conception et d'évaluation d'interfaces homme-machine..... 79**

2.1. Description des systèmes à la base de notre recherche.....	80
2.1.1. Système d'évaluation automatique d'interface homme-machine.....	80
2.1.2. Système de génération automatique de spécifications d'interface.....	84
2.1.3. Prototypage et système de spécification graphique d'interface.....	91
2.1.4. Approche d'interface "intelligente".....	97
2.1.5. Conclusion sur les systèmes mis en œuvre.....	103
2.2. Méthodes mises en place dans le cadre de notre recherche.....	104
2.2.1. Recensement de connaissances expertes.....	104
2.2.2. Positionnement des méthodes d'évaluation "statique" automatique et de génération automatique de spécifications d'interface dans la méthode en U.....	109
2.2.3. Méthode rapide d'analyse de système existant pour la spécification d'imagerie de même type.....	110
2.2.4. Méthode de création des bases du M.D.I.....	112
2.2.5. Méthode de développement d'imagerie axée sur le prototypage.....	113
2.2.6. Conclusion sur les méthodes mises en œuvre.....	115
Conclusion.....	116
Bibliographie du chapitre 2.....	117

## **Chapitre 3. Synthèse et perspectives de recherche autour du modèle $\nabla$ ..... 120**

3.1. Cycle de vie de système interactif et nouveau cadre théorique et méthodologique : le modèle $\nabla$ .....	121
3.2. Modèle $\nabla$ et réutilisabilité.....	123
3.2.1. Le principe de réutilisabilité.....	123
3.2.2. Réutilisabilité au niveau des interfaces homme-machine.....	124
3.2.3. Réutilisabilité au niveau des modules d'aide.....	125
3.2.4. Conclusion sur les perspectives en terme de réutilisabilité.....	127
3.3. Vers des méthodes basées sur le modèle $\nabla$ .....	127
3.4. Vers des outils logiciels et des ateliers de génie logiciel supportant le modèle $\nabla$ .....	129
3.5. Du Modèle $\nabla$ aux interfaces dites "intelligentes".....	130
Conclusion.....	132
Bibliographie du chapitre 3.....	133

## **Conclusion générale..... 135**

## **Première partie**

### *Curriculum vitae*



# Renseignements personnels

## Christophe KOLSKI

né le 15 Mars 1963 à Hénin-Liétard (62),  
de nationalité française, marié

## TITRES UNIVERSITAIRES

### Docteur en Automatique industrielle et humaine

*"Contribution à l'ergonomie de conception des interfaces graphiques homme-machine dans les procédés industriels : application au système expert SYNOP"*

Soutenue le 30 Janvier 1989 à l'Université de Valenciennes

Mention : très honorable

Directeur de la Thèse : Patrick MILLOT.

Composition du jury : Bernard FAVRE, Noël MALVACHE, Patrick MILLOT,  
René SOENEN, André TITLI et Pierre VIDAL

### Ingénieur EUDIL

Section Informatique, Mesures et Automatique

Université de LILLE I, Juin 1985, Mention : Assez Bien.

### DEUG A Sciences et Structures de la Matière

Université de LILLE I, Juin 1982, Mention : Assez Bien.

## AFFECTATIONS ACTUELLE ET ANTERIEURES

Depuis le 1/10/94

### Maître de Conférences en informatique 1ère classe

à l'institut des Sciences et Techniques de l'Université de Valenciennes et du Hainaut-Cambrésis (ISTV)

Bénéficiaire d'un contrat d'encadrement doctoral et de recherche

Du 1/10/90 au 31/9/94

### Maître de Conférences en informatique 2ème classe

à l'ISTV

Du 1/10/89 au 30/9/90

### Attaché temporaire d'enseignement et de recherche

à l'ISTV

Du 1/10/88 au 30/9/89

### Allocataire d'enseignement et de recherche

à l'IUT de Valenciennes

Du 1/10/85 au 30/9/88

### Allocataire de recherche

au Laboratoire d'Automatique Industrielle et Humaine

# Activités de recherche

## Thèmes de recherche :

Mes recherches portent depuis septembre 1985 sur l'ingénierie de conception et d'évaluation des interfaces graphiques homme-machine. Le domaine d'application est la supervision de systèmes industriels complexes. L'originalité de ces travaux réside dans la prise en compte explicite ou implicite de facteurs humains liés à l'interface homme-machine, qui se retrouvent dans la mise en oeuvre de nouvelles méthodes et structures logicielles pour la conception et l'évaluation.

Elles sont effectuées au sein du Laboratoire d'Automatique Industrielle et Humaine, partie du Laboratoire d'Automatique et de Mécanique Industrielles et Humaines (LAMIH), U.R.A. CNRS 1775, et particulièrement dans l'équipe "Informatique Industrielle et Communication Homme-Machine" dirigée par le Professeur Patrick MILLOT.

Ces recherches concernent particulièrement l'évaluation dite analytique d'interface homme-machine, la conception d'interface basée sur la génération automatique de spécifications, le prototypage d'imagerie, les interfaces dites "intelligentes" et la mise à disposition de concepteurs d'interfaces de connaissances livresques et didactiques. Ces recherches se basent sur des concepts et méthodes issus du génie logiciel, des sciences cognitives et de l'intelligence artificielle. Une des originalités de ces travaux réside dans la prise en compte explicite ou implicite de facteurs humains liés à l'interface homme-machine, qui se retrouvent dans la mise en oeuvre de nouvelles méthodes et structures logicielles pour la conception et l'évaluation.

Ces recherches sont effectuées dans un cadre méthodologique et théorique développé au laboratoire. Une présentation détaillée des différents thèmes, activités et perspectives de recherche est faite dans la suite de ce mémoire.

**Mots clés :** Interfaces homme-machine, interfaces "intelligentes", conception, évaluation, intelligence artificielle, génie logiciel, supervision de système industriel complexe, méthodologie de développement d'interface, système homme-machine, aide à la décision.

## Bilan quantitatif des travaux, ouvrages, articles et réalisations :

Encadrement de thèses	:	3	(+ 1 en cours)
Encadrement de D.E.A.	:	9	(+ 1 en cours)
Livre	:	1	
Revue	:	11	(+ 2 propositions)
Chapitre de livre	:	1	
Congrès internationaux	:	25	(+ 2 propositions acceptées)
Congrès national	:	1	
Conférences sur invitation	:	3	
Rapports de contrats	:	7	

# Encadrement de thésards et DEA

## Encadrement doctoral

### *Thèses passées :*

**MOUSSA Faouzi (1992).** *Contribution à la conception ergonomique des interfaces de supervision dans les procédés industriels : Application au système ERGO-CONCEPTOR.* Thèse de Doctorat, soutenue le 15 juillet 1992 à l'université de Valenciennes.

**TENDJAOUI Mustapha (1992).** *Contribution à la conception d'interface "intelligente" pour le contrôle de procédés industriels : application au Module Décisionnel d'Imagerie.* Thèse de Doctorat, soutenue le 20 novembre 1992 à l'université de Valenciennes.

**POULAIN Thierry (1994).** *Contribution du génie logiciel pour la conception et l'évaluation d'applications de supervision.* Thèse de Doctorat, soutenance le 25 février 1994 à l'université de Valenciennes.

### *Thèse en cours :*

**GRISLIN Martial (1994).** *Modèles formels pour le développement d'imagerie industrielle (titre provisoire).* Thèse de Doctorat, soutenance prévue pour décembre 1994.

## Encadrement au niveau DEA

### *DEA passés :*

**MOUSSA Faouzi (1988).** Réalisation de bases de connaissances en ergonomie de présentation d'information à l'écran, application au système expert SYNOP. Mémoire de DEA, juillet.

**BERRI Jawad (1990).** Formalisation de connaissances pour le module décisionnel d'imagerie. Mémoire de DEA, juillet.

**RAMAMONJISOA David (1990).** Contribution à l'étude et la réalisation d'une maquette d'un outil expert interactif ergonomique dans la démarche globale de conception d'interface graphique homme-machine. Mémoire de DEA, juillet.

**BEN HASSINE Taoufik (1991).** Contribution à l'étude et la mise en oeuvre d'une démarche de conception ergonomique d'interfaces homme-machine. Mémoire de DEA, juillet.

**LE STRUGEON Emmanuelle (1991).** Contribution à la conception et à la réalisation d'une interface évoluée : le Module Décisionnel d'Imagerie. Mémoire de DEA, juillet.

**CHACHOUA Mohamed (1992).** Contribution à la conception et la réalisation d'un atelier de création de synoptiques. Mémoire de DEA, juillet.

**VILAIN Bertrand (1992).** Contribution à la conception et à la réalisation d'un atelier de création de synoptiques. Mémoire de DEA, juillet.

**BELLOTTI Fabrice (1994).** Etude des modèles formels de conception d'interfaces homme-machine. Mémoire de DEA, juillet.

**MORVAN Michel (1994).** Conception d'une architecture de composants génériques pour la supervision de réseaux. Mémoire de DEA, septembre.

### *DEA en cours :*

**JOUGLET Michel (1995).** Développement objet d'imagerie de supervision et réutilisation de composants logiciels. Mémoire de DEA prévu pour juillet.

# Liste des travaux, ouvrages, articles et réalisations

En raison du caractère pluridisciplinaire de mes travaux, lié inévitablement aux aspects techniques et humains relatifs à la conception et à l'évaluation des interfaces homme-machine, ceux-ci ont fait l'objet de publications et de communications en :

- informatique, pour les aspects génie logiciel et intelligence artificielle,
- automatique, concernant ceux liés à la supervision de systèmes industriels complexes,
- sciences cognitives, pour les aspects liés au recensement et à la formalisation de connaissances ergonomiques et à la communication homme-machine.

## **I. LIVRE**

**KOLSKI C. (1993).** *Ingénierie des interfaces homme-machine, conception et évaluation*. Editions Hermès, Paris, 372 pages, août.

## **II. REVUES INTERNATIONALES AVEC COMITE DE LECTURE**

**KOLSKI C., MILLOT P. (1991).** *A rule-based approach to the ergonomic "static" evaluation of man-machine graphic interface in industrial processes*. International Journal of Man-Machine Studies, 35, pp. 657-674.

**TENDJAOUI M., KOLSKI C., MILLOT P. (1991).** *An approach towards the design of intelligent man-machine interfaces used in process control*. International Journal of Industrial Ergonomics, 8, pp. 345-361.

**MOUSSA F., KOLSKI C. (1991).** *ERGO-CONCEPTOR : système à base de connaissances ergonomiques pour la conception d'interface de contrôle de procédé industriel*. Technologies Avancées, vol. 2 (2), pp. 5-14.

**KOLSKI C., TENDJAOUI M., MILLOT P. (1992).** *A process method for the design of "intelligent" man-machine interfaces : case study : "the Decisional Module of Imagery"*. International Journal of Human Factors in Manufacturing, vol. 2 (2), pp. 155-175.

**KOLSKI C., WAROUX D. (1992).** *Une expérience de conception, de réalisation et d'évaluation par des conducteurs de leur propre synoptique de conduite de réseau gazier*. Le Travail Humain, tome 55, n°4, pp. 371-388.

**POULAIN T., GERME J.P., KOLSKI C. (1993).** *Un atelier de génie logiciel pour la spécification, la réalisation et l'évaluation de synoptiques industriels embarqués*. Génie Logiciel et Systèmes Experts, n° 31, pp. 58-70, juin.

**KOLSKI C., MILLOT P. (1993).** *Problems in telemaintenance and decision aid criteria for telemaintenance system design*. International Journal of Industrial Ergonomics, 11, pp. 99-106.

**KOLSKI C., DURIBREUX M. (1993).** *Towards explicit consideration of time in cognitive engineering workbenches for "intelligent" assistance system design*. International Journal of Industrial Ergonomics, 11, pp. 125-134.

**KOLSKI C., LE STRUGEON E., TENDJAOUI M. (1993).** *Implementation of AI techniques for "intelligent" interface development*. Engineering Application of Artificial Intelligence, vol. 6 (4), pp. 295-305.

**VILAIN B., POULAIN T., KOLSKI C.** *Un atelier logiciel de spécification, réalisation et évaluation de synoptiques industriels embarqués*. En cours d'édition dans la revue : Automatique-Productique-Informatique Industrielle (APII).

**GRISLIN M., KOLSKI C. (en soumission)** *Evaluation des interfaces homme-machine lors du développement de système interactif : revue de synthèse*. Proposé à la revue Technique et Science Informatiques (TSI) en juillet 1994.

**KOLSKI C., TENDJAOUI M. (en soumission)** *Architectures d'interfaces homme-machine dites "intelligentes". Partie I : du modèle de SEEHEIM aux concepts d'interfaces "intelligentes". Partie II : vers une approche d'interface "intelligente" : le module décisionnel d'imagerie*. Proposé à la revue d'Intelligence Artificielle en juillet 1994.

### **III. REVUE NATIONALE SANS COMITE DE LECTURE**

**KOLSKI C. (1993).** *Ergonomie de la supervision, conception d'interfaces homme-machine*. AXES Robotique-Automatique, n°70/71, décembre 92, janvier 93.

### **IV. CHAPITRE DE LIVRE**

**KOLSKI C., MILLOT P. (1992).** *Potential contribution of new hypermedia technologies for man-machine communication in industrial processes*. "Hypermedia Courseware: Structures of Communication and Intelligent Help", A. Oliveira (Ed.), NATO ASI Series, Springer-Verlag.

### **V. CONGRES INTERNATIONAUX AVEC ACTES ET COMITE DE LECTURE**

**BINOT C., KOLSKI C., ROGER D., WILLAEYS D. (1987).** *Graphic information evaluation and expert system in ergonomic conception of control views*. IECON'87 thirteenth annual IEEE Industrial Electronics Society Conference, Cambridge, Massachusetts, USA, 1-6 November.

**KOLSKI C., BINOT C., MILLOT P., ROGER D. (1988).** *Use of ergonomic concepts by an expert system for man machine graphic interface optimization*. PROLAMAT'88, IFIP/IFAC 7th International Conference Software for Manufacturing, Dresden, DDR, 14-17 Juny. Publié dans un livre : "Software for manufacturing", D. Kochan and G. Olling (Eds.), North-Holland, pp. 559-567, 1989.

**KOLSKI C., VAN DAELE A., MILLOT P., DE KEYSER V. (1988).** *Towards an intelligent editor of industrial control views, using rules for ergonomic design*. IFAC Workshop "Artificial intelligence in real-time control", Clyne Castle, Swansea, Great Britain, 21-23 September.

**KOLSKI C., MILLOT P. (1989).** *Démarches ergonomiques d'évaluation et de conception d'interfaces graphiques homme-machine à l'aide de techniques d'intelligence artificielle : évolutions vis-à-vis de l'ergonomie*. XXVème Congrès de la SELF : Evolutions technologiques et Ergonomie, Lyon, 4-6 Octobre.

**KOLSKI C., MILLOT P. (1990).** *Potential contribution of new hypermedia technologies for man-machine communication in industrial processes*. NATO Advanced research Workshop "Structures of communication and intelligent Help for Hypermedia Courseware", Espinho, Portugal, 19-24 April. Publié dans un chapitre de livre : cf. IV

**MOUSSA F., KOLSKI C., MILLOT P. (1990).** *Etude de l'intégration d'un outil expert interactif ergonomique dans la démarche globale de conception d'interface graphique homme-machine*. Colloque International "l'Ordinateur, l'Homme et l'Organisation II", Nivelles, Belgique, 9-11 Mai.

**KOLSKI C., TENDJAOUI M., MILLOT P. (1990).** *An "intelligent" interface approach*. The second International Conference on "Human aspects of advanced manufacturing and hybrid automation", Honolulu, Hawaii, USA, August 12-16. Publié dans un livre : "Ergonomics of hybrid automated systems II", W. Karwowski and M. Rahimi (Eds.), Elsevier, pp. 587-594, 1990.

**MOUSSA F., KOLSKI C., MILLOT P. (1990).** *Artificial intelligence approach for the creation and the ergonomic design of man-machine interfaces in control room*. Ninth European Annual Conference on "Human decision making and manual control", Varese, Italy, September 10-12.

**TENDJAOUI M., KOLSKI C., MILLOT P. (1990).** *Interaction between real-time aid expert system, intelligent interface and human operator*. International Symposium Computational Intelligence 90 "Heterogeneous knowledge representation systems", September 24-28, 1990, Milano, Italy.

- KOLSKI C., MOUSSA F., MILLOT P. (1990).** *Formalisation de connaissances ergonomiques dans le domaine de l'ergonomie de conception d'interfaces graphiques Homme-Machine.* XXVI Congrès de la SELF "Méthodologie et outils d'intervention et de recherche en ergonomie", Montréal, Canada, 3-6 Octobre.
- GAMBIEZ F., KOLSKI C., TANG X. MILLOT P. (1991).** *Man-machine interface for PREDEX, an expert system shell for process control and supervision.* IMACS International Workshop "Decision support systems, Qualitative reasoning", Toulouse, March 13-15. Publié dans un livre : "Decision support systems and qualitative reasoning", M.G. Singh and L. Travé-Massuyès (Eds.), North-Holland, pp. 341-346, 1991.
- MANDIAU R., KOLSKI C., MILLOT P. (1991).** *Perspective of Human-Machine system modelling by logical approach : Contribution of technics issued from Distributed Problem Solving.* 11th Congress of the International Ergonomics Association. July 15-20, Paris. Publié dans un livre : "Designing for Everyone vol. 1", Y. Queinnec and F. Daniellou (Eds.), Taylor & Francis, pp. 436-438, 1991.
- TENDJAOUI M., KOLSKI C., GAMBIEZ F., MILLOT P. (1991).** *Towards an "intelligent" interface between decision aid tool and operator in control rooms.* 11th Congress of the International Ergonomics Association. July 15-20, Paris. Publié dans un livre : "Designing for Everyone vol. 1", Y. Queinnec and F. Daniellou (Eds.), Taylor & Francis, pp. 652-654, 1991.
- MANDIAU R., KOLSKI C., MILLOT P., CHAIB-DRAA B. (1991).** *A new approach for the cooperation between human(s) and assistance system(s) : a system based on intentional states.* World Congress on Expert Systems, Orlando, Florida, Dec. 16-19.
- TENDJAOUI M., KOLSKI C., MILLOT P. (1991).** *Knowledge based interface approach for real-time aid expert system.* IFAC/IMACS "SAFEPROCESS'91" Symposium, 10-13 September, Baden-Baden, Germany.
- POULAIN T., KOLSKI C., MILLOT P. (1991).** *Contribution of the prototyping techniques for the ergonomic design of process control synopsis.* Tenth European Annual Conference on "Human Decision Making and Manual Control", Liège, Belgium, 11-13 November.
- KOLSKI C., MOUSSA F. (1991).** *Une approche d'intégration de connaissances ergonomiques dans un atelier logiciel de création d'interfaces pour le contrôle de procédé.* Quatrièmes Journées Internationales : Le génie logiciel et ses applications, Toulouse, Editions EC2, 9-13 décembre.
- POULAIN T., KOLSKI C. (1992).** *Monde réel et monde virtuel, la problématique du contrôle de procédé par un opérateur humain.* Conférence Informatique 92 "L'interface des mondes réels et virtuels", Montpellier, 23-27 Mars.
- KOLSKI C. (1992).** *Formalization approaches of ergonomic knowledge for "intelligent" design, evaluation and management of man-machine interface in process control.* IFAC Symposium on Intelligent Components and Instruments for Control Applications, Malaga, Spain, May 20-22.
- MOUSSA F., KOLSKI C., MILLOT P. (1992).** *A formal methodology for ergonomic design of Man-Machine interfaces.* 5th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Man-Machine Systems, The Hague, The Netherlands, June 9-11.
- KOLSKI C., MILLOT P. (1992).** *Decision aid criteria to integrate a telemaintenance tool into the maintenance man-machine system.* The Annual International Industrial Ergonomics and Safety Conference '92, June 10-14, Holiday Inn, Denver, Colorado, USA. Publié dans un livre : "Advances in Industrial Ergonomics and Safety IV", S. Kumar (Ed.), Taylor & Francis, pp. 51-58, 1992.
- KOLSKI C., DURIBREUX M. (1992).** *Towards cognitive engineering workbenches considering the time for the "intelligent" design of operator assistance tools.* The Annual International Industrial Ergonomics and Safety Conference '92, June 10-14, Holiday Inn, Denver, Colorado, USA. Publié dans un livre : "Advances in Industrial Ergonomics and Safety IV", S. Kumar (Ed.), Taylor & Francis, pp. 1063-1070, 1992.  
A l'occasion de ce congrès, j'étais co-chairman dans la session "Cognitive ergonomics".
- LE STRUGEON E., TENDJAOUI M., KOLSKI C. (1992).** *Knowledge specification and representation for an "intelligent" interface devoted to process monitoring and supervision.* IFAC/IFIP/IMACS International Symposium on Artificial Intelligence in real-time control, Delft, The Netherlands, June 16-18.
- MOUSSA F., KOLSKI C. (1992).** *Vers une formalisation d'une démarche de conception de synoptiques industriels : application au système ERGO-CONCEPTOR.* Colloque ERGO-IA "Ergonomie et Informatique Avancée", 7-9 Octobre, Biarritz.

**GRISLIN M., KOLSKI C., ANGUE J.C. (1993).** *Towards an organization of man-machine interface evaluation techniques by a usability criteria grid.* XII European Annual Conference on Human Decision Making and Manual Control, Kassel, 22-24 June.

**DURIBREUX M., KOLSKI C., HOURIEZ B. (1995).** *Towards an integration of cognitive ergonomics concepts in knowledge based system development methodologies.* Proposition acceptée pour le 6th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Man-Machine Systems, M.I.T., Cambridge, USA, June 27-29.

**GRISLIN M., KOLSKI C., ANGUE J.C. (1995).** *Human-computer interface evaluation in industrial complex systems : a review of usable techniques.* Proposition acceptée pour le 6th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Man-Machine Systems, M.I.T., Cambridge, June 27-29.

## **VI. CONGRES NATIONAL**

**KOLSKI C., GRZESIAK F., MILLOT P. (1988).** *Outil de conception et d'évaluation de la page-écran utilisant la technologie des systèmes experts.* Congrès : Systèmes Experts et Télématique, Paris, Editions EC2, 28-29 Janvier.

## **VII. CONFERENCES SUR INVITATION**

**KOLSKI C. (1992).** *Ergonomie de la supervision, conception d'interfaces homme-machine.* Journée "La Supervision", Centre des Technologies Nouvelles, Caen, 19 Mars.

**KOLSKI C. (1993).** *Communication homme-machine : nouveaux développements.* 2ème Symposium Franco-Chinois sur la recherche et la conception en ergonomie, Paris, 23-26 novembre.

**KOLSKI C. (1994).** *Ingénierie des interfaces homme-machine.* Séminaire COMETT 94 "Conception et ergonomie des interfaces homme-machine, état de l'art et perspectives industrielles", Institut International pour l'Intelligence Artificielle, Compiègne, 21 juin.

## **VIII. RAPPORTS DE CONTRATS**

**KOLSKI C., MOUSSA F., MILLOT P., (1988 et 89).** *Mise en oeuvre d'une base de connaissances ergonomiques exploitables par le système expert SYNOP pour l'aide à la conception d'images embarquées en automobile et évaluation.* Rapports établis dans le cadre de la convention RENAULT N°UV/14, LAIH, Université de Valenciennes, Juin 1988 et Février 1989.

**GAMBIEZ F., TENDJAOUI M., KOLSKI C., MILLOT P. (1990).** *Etude et mise en oeuvre d'un module d'interface pour l'outil PREDEX de développement de système expert de filtrage et de gestion d'alarmes pour les procédés continus.* Rapport final établi dans le cadre de la convention LAIH/S2O, Juin.

**KOLSKI C. (1991).** *Assistance graphique à l'opérateur : concepts de base et recommandations ergonomiques.* Support d'intervention établi dans le cadre de la collaboration GAZ DE FRANCE/LAIH, septembre.

**KOLSKI C., MILLOT P. (1992).** *Télémaintenance de systèmes automatisés, guide ergonomique de conception.* Rapport établi dans le cadre de la convention L.A.I.H./B+ Development, Rapport final, Mai.

**KOLSKI C. (1992).** *Conception ergonomique d'interface opérateur en salle de contrôle : lignes directrices.* Guide de conception rédigé dans le cadre de la convention L.A.I.H./SOLVAY, Janvier.

**POULAIN T., VILAIN B., KOLSKI C., MILLOT P. (1991, 92 et 93).** *Atelier de création de synoptiques industriels.* Rapports de contrat dans le cadre d'une collaboration entre CSEE, 3IP et le LAIH faisant suite à un appel d'offres de l'ANVAR, décembre 1991, mai 1992, juin 1993.

**KOLSKI C., RIERA B., BERGER T., MILLOT P. (1993).** *Analyse en salle de contrôle du procédé H2O2 à JEMEPPE, lignes directrices pour la conception d'imagerie pour un procédé de même type.* Rapport final établi dans le cadre de la convention L.A.I.H./SOLVAY, septembre.

# Activités d'enseignement

## INFORMATIQUE

### **Génie logiciel**

- 16 h Cours et 14 h TD en Maîtrise d'informatique depuis Octobre 93

Contenu : Les principaux cycles de vie issus du génie logiciel sont détaillés en insistant sur la problématique et les besoins de chaque étape ; plusieurs méthodes d'analyse, de spécification et de conception sont expliquées. Une part importante du cours est consacrée à l'orientation objet.

### **Intelligence artificielle et systèmes experts**

- 20 h Cours en Magistère Audiovisuel depuis Octobre 90

- *Activité passée : 20 h Cours en 91-92 et 92-93 en DETS Management de la Production*

Contenu : Dans une première partie, j'explique les notions principales liées aux systèmes experts dits de première génération. Dans une seconde partie, je parcours des notions rattachées aux systèmes à base de connaissance dits de seconde génération.

### **Ingénierie de la connaissance**

- 12 h Cours en DESS Informatisation et Communication Homme-Machine depuis Octobre 89

Contenu : J'explique d'abord les problèmes liés à l'ingénierie de la connaissance. Puis, je présente des méthodes et techniques d'acquisition de connaissance, de même que les aspects méthodologiques liés au développement de systèmes à base de connaissance.

### **Langage LISP**

- 12 h TD et 24 h TP en DESS Productique depuis Octobre 87 (3ème année EIGIP depuis 93-94)

- 12 h Cours et 16 h TP DESS ICHM depuis Octobre 86

- *Activité passée : 10 h Cours, 4 h TD et 8 h TD en DU ISHM en 88-89 et 89-90*

Contenu : J'explique les fondements du langage LISP, et les fonctionnalités mises à la disposition du développeur. Les étudiants doivent en TP réaliser un environnement minimal de système à base de connaissance avec ses interfaces expert et utilisateur. Notons que les étudiants de l'EIGIP doivent, lors de séances supplémentaires, réaliser un outre un système à base de blackboard.

### **Langage PROLOG**

- *Activité passée : 10 h Cours et 8 h TP en DESS ICHM en 86-87 et de 89-90 à 92-93*

Contenu : J'expliquais les fondements du langage, et les fonctionnalités mises à la disposition du développeur. La première partie du TP portait sur la recherche d'un chemin optimal dans un circuit, alors que la seconde consistait en la réalisation d'un système à base de connaissance utilisable dans une agence matrimoniale.

### **Langage SMALLTALK**

- *Activité passée : 6 h Cours et 8 h TP en DESS ICHM en 88-89*

Contenu : Après une première partie consacrée aux concepts de la programmation orientée objets, j'expliquais les fondements du langage, ainsi que les fonctionnalités mises à la disposition du développeur. Le TP consistait en la mise en oeuvre de classes de compteurs.

### **Algorithmique**

- *Activité passée : 40 h TD et 40 TP en DEUG A 1ère année de 88-89 à 93-94*

Contenu : Les thèmes traités étaient variés : types standards et complexes, programmation modulaire, algorithmiques numérique et non numérique, relations de récurrence, manipulation de tableaux et de fichiers, etc.



## **INTERFACES HOMME-MACHINE**

### **Ergonomie des logiciels**

- 10 h Cours en Magistère Audiovisuel depuis Octobre 90

Contenu : De nombreux concepts liés aux utilisateurs d'un système informatique sont d'abord décrits. Ensuite, des méthodes permettant de prendre en compte ceux-ci tout au long du cycle de développement sont expliquées.

### **Méthodes et modèles de conception d'une application interactive**

- 8 h Cours à l'IUP GEII depuis Octobre 93

Contenu : Des modèles d'architectures d'applications interactives sont d'abord passés en revue. Puis, des méthodes de conception des interfaces homme-machine sont expliquées.

### **Conception d'imagerie de supervision**

- 8 h Cours et 8 h TD à l'IUP GEII depuis Octobre 93

- *Activité passée : 24 h Cours et 8 h TD en DU ISHM en 88-89 et 89-90*

Contenu : Des modes de présentation de l'information sur écran sont d'abord disséqués et critiqués. Lors d'études de cas, il s'agit ensuite pour les étudiants de concevoir des imageries de supervision de systèmes industriels complexes.

### **Mise en oeuvre de générateur d'imagerie, exemple du progiciel IMAGIN**

- *Activité passée : 8 h TP en DESS Productique en 89-90*

Contenu : Le TP a porté sur la génération des parties statiques et dynamiques d'une imagerie de supervision de système complexe.

## **Activités antérieures au doctorat**

## **AUTOMATIQUE ET INFORMATIQUE INDUSTRIELLE**

### **Programmation d'automates industriels**

- *Activité passée : 40 TP à l'IUT Maintenance Industrielle en 85-86*

Contenu : Mes premières activités d'enseignement ont porté sur la programmation des automates industriels (MODICON, TSX-80, TSX-47, PB-100 et TERGANE-80).

### **GRAFCET et analyse de la partie commande**

- *Activité passée : 12 h TD au CAPET de Mécanique en 88-89*

Contenu : Après une présentation du GRAFCET, ce cours étudiait particulièrement les notions de cycle de production, de sûreté de fonctionnement et de diagramme des tâches.

### **Asservissement et régulation de processus**

- *Activité passée : 78 h TP à l'IUT GEII en 88-89*

Contenu : En poste d'allocataire d'enseignement et de recherche à l'IUT GEII, j'ai donné des TP consistant en l'étude de systèmes linéaires et non linéaires, de la notion de stabilité de système, de techniques d'asservissement, de régulation et d'identification de processus.

### **Microprocesseur**

- *Activité passée : 66 h TP à l'IUT GEII en 88-89*

Contenu : Ce TP consistait en fait en un projet sur 66 heures de réalisation d'une serrure électronique utilisant un monochip 8051. La programmation était réalisée en Assembleur au moyen d'un outil de développement ou d'un PC associé à un émulateur.

# Activités d'administration et autres responsabilités collectives

## Activités administratives

Depuis octobre 1991, je suis le correspondant du laboratoire auprès de la bibliothèque universitaire et ai la charge de centraliser les demandes d'abonnements.

*Activités passées :* De 1986 à 1989, j'étais membre étudiant du conseil des Sports de l'Université de Valenciennes et du Hainaut-Cambrésis. De 1990 à 1992, j'étais correspondant du laboratoire auprès du service des Sports.

De 1986 à 1988, j'étais membre élu du conseil scientifique de l'institut des sciences et techniques de l'Université de Valenciennes et du Hainaut-Cambrésis (étudiant 3ème cycle).

De 1990 à fin 1993, j'ai fait partie du Conseil Scientifique de l'Unité de Recherche en Informatique et Automatique Humaine, n° 1118, associée au C.N.R.S.

## Responsabilités pédagogiques

Depuis octobre 1989, j'ai la responsabilité d'organiser la répartition des T.P. en *DESS Informatisation et Communication Homme-Machine*.

Depuis octobre 1992, je suis responsable des stages à *l'IUP Génie Electrique et Informatique Industrielle option Automatisation Intégrée*. En 1992-93, 25 étudiants ont été placés et suivis en stage. Ce nombre est monté à 40 en 1993-94, puis 42 en 1994-95.

Depuis octobre 1993, je suis responsable des stages en *Maitrise d'informatique*. J'ai eu la charge de 10 étudiants en 1993-94. Ce nombre est monté à 17 en 1994-95.

Depuis octobre 1993, je fais partie du conseil de perfectionnement de *l'IUP Génie Electrique et Informatique Industrielle*.

Je suis tuteur pédagogique en informatique d'Emmanuelle Le Strugeon (depuis octobre 1991) et de Patrice Caulier (depuis octobre 1992), en thèse tous les deux dans l'équipe dirigée par le professeur Patrick MILLOT.

*Activité passée :* En 1988-89, j'ai organisé la répartition des T.P. d'Intelligence Artificielle en *DU Informatique et Système Homme-Machine*.

## **Seconde partie**

*Méthodes et modèles de conception et  
d'évaluation des interfaces homme-machine*

# Introduction générale

L'importance accordée aux interactions homme-machine en général est actuellement énorme. Pour s'en convaincre, il suffit de constater le nombre de congrès, d'ouvrages et de revues qui leur sont consacrées. L'orientation prise par les chercheurs et les industriels pour couvrir ce domaine est désormais résolument pluridisciplinaire.<sup>1</sup>

Ce mémoire d'habilitation se situe dans cette mouvance pluridisciplinaire en étant consacré à l'étude de méthodes de conception et d'évaluation des interfaces homme-machine dans les systèmes complexes. Ce domaine de recherche se positionne au carrefour du génie logiciel, de l'intelligence artificielle et des sciences cognitives.

Toutefois, il s'agit de souligner que la problématique, les modèles, les outils et les méthodes utilisés dépendront en grande partie du champ d'application visé (par exemple la bureautique, les systèmes d'information, les systèmes grand publics, les systèmes industriels complexes, les jeux vidéo, la conception assistée par ordinateur, l'enseignement assisté par ordinateur, etc). Pour notre part, nous nous sommes essentiellement focalisés sur un domaine : *les interfaces graphiques homme-machine utilisées dans des salles de supervision pour surveiller et commander des systèmes complexes* (chimiques, nucléaires, réseaux d'énergie, transports...). Ces interfaces offrent un vaste champ d'étude de par la complexité des tâches confiées à l'homme, par la diversité des critères (liés à la production, la sécurité, l'économie, la qualité, l'écologie, etc) nécessaires pour mener à bien ces tâches, ainsi que par la variété des aides informatiques possibles à apporter à l'homme.

## *L'homme*<sup>2</sup>

En effet, suite à l'évolution technologique, l'être humain en salle de supervision doit de plus en plus réaliser des tâches mentales complexes de résolution de problèmes, au moyen d'outils graphiques, et ceci lors de situations dynamiques, aux exigences bien particulières (SHERIDAN, 1988 ; VAN DAELE, 1992 ; DE KEYSER et al., 1992).

L'interface homme-machine joue donc un rôle essentiel dans l'efficacité et la fiabilité du système. La figure 1 montre un utilisateur d'une interface homme-machine dans une salle de supervision d'un système complexe : déconnecté physiquement de celui-ci, on le voit en train de résoudre un problème, impliquant de nombreuses variables, à l'aide en particulier des outils graphiques mis à sa disposition dans la salle.

<sup>1</sup> Voir à ce sujet le contenu d'un enseignement type des interactions homme-machine proposé par le groupe SIGCHI (Special Interest Group on Computer-Human Interaction) de l'ACM (Association for Computing Machinery), association regroupant plusieurs dizaines de milliers d'informaticiens dans le monde entier (ACM SIGCHI, 1992).

<sup>2</sup> Bien entendu, le terme "homme" doit être pris dans un sens général et est ici synonyme d'être humain. Notons que le mot "man" disparaît aux Etats-Unis des expressions relatives aux interactions homme-machine, au profit de "human" ou "user". Par exemple, la célèbre revue "International Journal of Man-Machine Studies" a changé de nom en 1994 pour devenir "International Journal of Human-Computer Studies". Par commodité, le terme "homme" sera utilisé dans ce mémoire.



Figure 1 : Exemple de salle de supervision (Photothèque DETN - GAZ de FRANCE)

### *Les tâches humaines*

Exigeant un haut niveau de connaissance, les tâches humaines dans les systèmes complexes ont été regroupées par ROUSE (1983) en quatre grandes classes : <sup>3</sup>

- *Les tâches de transition* : celles-ci correspondent aux changements de régime du système (arrêt, démarrage, changement de points de fonctionnement). L'interface homme-machine doit alors permettre d'effectuer des procédures pré-établies, de juger de leur bonne répercussion sur le système et de surveiller constamment la bonne évolution de celui-ci.
- *Les tâches de contrôle et de suivi de l'installation* : l'interface homme-machine doit faciliter à l'homme la surveillance de l'état du système, de façon à détecter et à anticiper l'apparition d'un événement anormal, et optimiser, par des réglages fins, la production.
- *Les tâches de détection de défaut et de diagnostic* : par l'apparition d'alarmes ou/et la possibilité d'observer une évolution anormale de certaines variables du système, l'homme doit pouvoir détecter les défauts et effectuer son diagnostic en conséquence. Il est alors nécessaire à l'interface homme-machine de permettre une mise en évidence aisée des relations cause/effet entre les variables du système.
- *Les tâches de compensation ou de correction* : afin de rétablir le fonctionnement normal du système, l'interface doit aider l'être humain à décider des actions à accomplir et à visualiser leurs effets sur le système.

### *L'interface homme-machine*

Pour réaliser ces tâches complexes mettant en jeu parfois des milliers de variables, l'homme dispose d'interfaces homme-machine lui présentant graphiquement un ensemble d'informations. Ces interfaces peuvent être couplées à des modules d'assistance, aux raisonnements qualifiés parfois "d'intelligents" s'ils

<sup>3</sup> En plus de ces quatre classes de tâches humaines, DE KEYSER et al. (1987) insistent particulièrement sur l'aspect collectif du travail. En effet, dans certaines situations inhérentes aux tâches citées, l'être humain doit organiser une distribution optimale des actions dans le temps. Il a alors à gérer et à synchroniser des actions séquentielles et/ou parallèles, effectuées éventuellement par d'autres intervenants. Ces tâches doivent se baser sur des estimations du moment et de la durée des actions.

sont basés sur des techniques d'intelligence artificielle, ou typiquement algorithmiques. Ces modules d'assistance peuvent viser l'aide au diagnostic, à la prédiction de défauts, au filtrage d'alarmes, etc (MILLOT, 1988 ; TABORIN, 1989).

Dans ce mémoire, nous nous intéressons à l'interface homme-machine selon le schéma de principe visible en figure 2 : celle-ci est connectée à l'application, constituée du système industriel complexe, et d'éventuels modules d'aide à la décision. Trois remarques importantes peuvent être formulées :

- Des flèches n'ont pas été placées sur les connections entre l'interface homme-machine et les composants de l'application, puisque les flux d'information peuvent varier selon l'application visée. Le lecteur se référera à ce sujet à la thèse de TENDJAOUI (1992) qui propose une classification des modules d'assistance en discutant de tous les flux d'information possibles entre les composants du système homme-machine.
- Plusieurs personnes peuvent avoir à travailler en équipe et utiliser plusieurs interfaces homme-machine connectées à des modules d'assistance différents. Même si les notions de travail collectif (ou coopératif ou collaboratif) font l'objet de recherches actives actuellement, nous resterons dans un cadre général. C'est pourquoi les termes "utilisateur" et "interface homme-machine" seront utilisés au singulier dans la suite du mémoire.
- A partir de ce schéma de principe, il sera possible dans ce mémoire de mieux situer certains modèles d'interface connus, ou encore des notions d'interfaces dites évoluées.

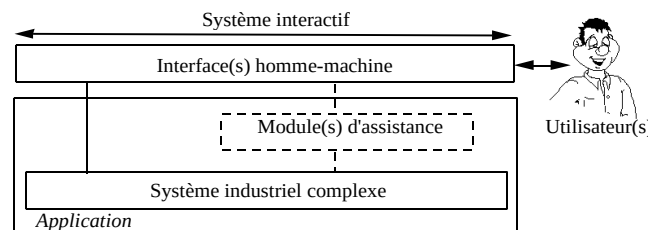


Figure 2 : Positionnement des interfaces homme-machine considérées dans ce mémoire

### *Le contenu des vues*

Les interfaces homme-machine doivent synthétiser l'état de l'application et aider l'être humain lors de ses activités. Celui-ci ne possède donc pas de vue directe sur le système et sur le résultat des actions, et doit donc se forger une représentation mentale du fonctionnement à distance du système et des différentes variables qui le composent. Il travaille alors à travers la représentation mentale qu'il se fait à chaque instant du système, représentation qu'il obtient grâce notamment à l'interface homme-machine (NORMAN, 1986 ; RASMUSSEN, 1986 ; RICHARD et al., 1990).

L'interface est centralisée dans ce but sur un ou plusieurs écrans de visualisation, et est constituée d'ensembles de vues, parfois plusieurs centaines ou milliers selon la complexité du système, et dont deux exemples sont visibles en figure 1. Chaque vue graphique peut présenter plusieurs dizaines d'informations différentes. Une vue est composée de deux parties distinctes : (i) la partie statique (ou fond de plan) constituée d'informations non modifiables, restant en permanence à l'écran, (ii) la partie dynamique constituée d'informations animées permettant de suivre les évolutions quantitatives et qualitatives des variables du système.

Les progiciels de création et d'animation d'imageries industrielles, actuellement disponibles sur le marché, permettent plus ou moins facilement la création de la partie "statique" à l'aide de fonctions graphiques de base permettant de dessiner des rectangles, des vecteurs, du texte, des arcs de cercles, etc. Pour la partie "dynamique", ces progiciels proposent différents types de fonctions plus évoluées, par

exemple des symboles, des messages, des barre-graphes, des cadrans, des courbes, des compteurs, etc. Le lecteur se référera au chapitre 5 de l'ouvrage de KOLSKI (1993) qui recense la plupart des modes de représentation traditionnels ou spécifiques que l'on peut retrouver dans les salles de supervision de systèmes complexes.

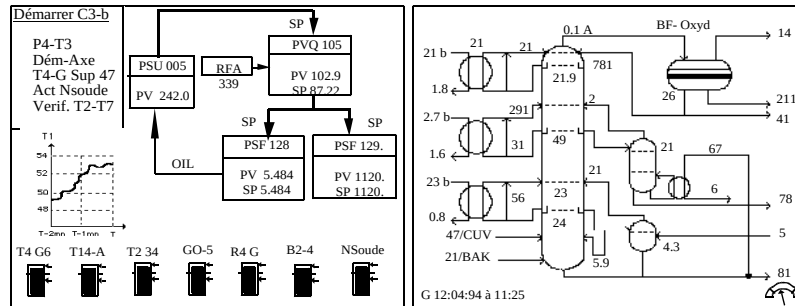


Figure 1 : Deux exemples de vues graphiques utilisées dans les systèmes complexes

### *L'enjeu*

En raison du nombre élevé de variables à considérer et de la complexité des tâches à réaliser, il s'avère que l'utilisation des interfaces homme-machine peut entraîner des erreurs humaines aux conséquences parfois catastrophiques, aussi bien du point de vue humain, qu'économique ou écologique (Three Miles Island, Tchernobyl, crashes d'avion de ligne ou de chasse, etc). Il existe à ce sujet dans la littérature de nombreuses classifications et études de l'erreur humaine, telles celles de ROUSE et ROUSE (1983), LEPLAT (1985), CELLIER (1990) ou REASON (1990).

Ces erreurs proviennent souvent de lacunes lors de l'analyse des besoins informationnels pour les différentes tâches à accomplir, de connaissances insuffisantes de la part des concepteurs des ressources et limites des utilisateurs, d'erreurs de la part des concepteurs dans la présentation et la structuration de l'information sur les différents écrans, d'une définition inadaptée de l'architecture de l'interface homme-machine, et/ou d'insuffisances lors des évaluations en situations réelle ou simulée.

Concernant les interfaces, il s'agit d'éviter de telles erreurs en prenant le problème à la base, c'est-à-dire au niveau de leur développement. Dans la mesure où des outils graphiques de plus en plus puissants apparaissent actuellement sur le marché (COUTAZ, 1990, MEINADIER, 1991), ce n'est pas sur la réalisation effective des interfaces que portera notre réflexion. C'est plutôt sur des méthodes de conception des interfaces, visant à intégrer plus explicitement les facteurs humains que nous insisterons.

### *Le contenu de ce mémoire*

Dans les systèmes complexes, les facteurs humains sont trop souvent négligés voire ignorés par l'équipe chargée du développement des applications interactives, par manque de méthodes ou de modèles. L'objectif des travaux rapportés dans ce mémoire est donc de contribuer à une prise en compte plus explicite des facteurs humains lors du développement. Les aspects méthodologiques du développement serviront donc de canevas théorique tout au long de ce mémoire qui est organisé en trois chapitres :

- Le premier est consacré à un état de l'art sur le domaine de recherche. Après une étude critique des différents cycles de développement du génie logiciel, il recensera autour d'un cadre théorique et méthodologique un ensemble de méthodes, modèles, techniques et outils contribuant au développement d'applications interactives dans les systèmes complexes.
- Le second chapitre sera bâti autour d'un ensemble de contributions à l'ingénierie de conception et d'évaluation des interfaces homme-machine. Ces contributions résultent d'un travail d'équipe

depuis mon arrivée au laboratoire en septembre 1985. Après une présentation des logiciels réalisés dans le cadre de cette recherche, ce chapitre présentera différentes approches d'acquisition, de formalisation et d'exploitation de connaissances. Par la suite, plusieurs méthodes de conception ou évaluation d'interfaces seront décrites.

- Enfin, le troisième chapitre tirera les leçons des aspects méthodologiques étudiés dans les deux premiers en proposant un modèle de développement d'applications interactives. Au cours de ces prochaines années, l'approfondissement de ce modèle, conciliant génie logiciel et aspects cognitifs, devrait servir de cadre de pensée pour nos recherches.



# Chapitre 1

## *Etat de l'art sur le domaine de recherche*

Ce premier chapitre a pour objet le développement d'applications interactives dans lesquelles nous constaterons que les méthodes du génie logiciel s'avèrent aujourd'hui insuffisantes malgré un essor considérable, ces dernières années, des outils de spécification et de réalisation des interfaces homme-machine. Il est donc nécessaire d'enrichir progressivement ces méthodes afin de les adapter à la problématique, liée aux interactions entre l'homme et la machine, dans les systèmes complexes.

Actuellement, de l'ensemble des sciences techniques et humaines émergent des outils, des techniques, des méthodes et des modèles susceptibles de contribuer au développement des interfaces homme-machine. Dans les systèmes complexes, le développement peut et doit en conséquence se baser sur une démarche pluridisciplinaire utilisant conjointement les spécificités :

- *des l'informatique pour les modèles, les méthodes et les techniques de développement d'outils logiciels* : On distingue d'abord plusieurs cycles de vie des systèmes et des logiciels, ainsi qu'un ensemble de méthodes d'analyse et de modélisation. De plus, l'intelligence artificielle fournit des modèles et des techniques d'acquisition, de formalisation et d'exploitation des connaissances. L'informatique graphique apporte de nombreux outils, méthodes et modèles liés aux aspects graphiques.
- *des sciences cognitives pour les facteurs humains* : elles apportent des méthodes d'évaluation, d'analyse des tâches et des activités des utilisateurs, des connaissances sur l'homme, sur les systèmes socio-techniques, des recommandations visant l'amélioration des conditions de travail en général et de l'ergonomie des logiciels en particulier.

Dans le but de recenser les méthodes, modèles, techniques et outils susceptibles de contribuer au développement des interfaces homme-machine, ce chapitre se compose de quatre parties.

- Une interface homme-machine constitue avant tout une application informatique. C'est pourquoi il s'agit dans la première partie de recenser les différents modèles et méthodes de développement issus du génie logiciel. Après une étude critique de ceux-ci vis-à-vis du développement de logiciels fortement interactifs, il importera de se définir un cadre théorique et méthodologique.
- Dans le développement d'interfaces homme-machine pour les systèmes complexes, il s'agit de considérer que l'utilisateur doit effectuer des tâches hautement cognitives de résolution de problème dans des situations très différentes. Il s'avère donc que l'analyse et la modélisation de l'utilisateur et des tâches humaines auront une importance énorme, nécessitant l'utilisation de méthodes rigoureuses. C'est pourquoi la deuxième partie recense de telles méthodes.
- La troisième partie présente plusieurs modèles d'architectures d'interface homme-machine, pour la plupart dérivées du célèbre modèle de SEEHEIM. Puis, progressivement, elle décrit des concepts et architectures d'interfaces dites évoluées ou "intelligentes".
- Enfin, en raison de l'apparition d'outils graphiques de plus en plus puissants facilitant une démarche de conception itérative, la spécification, la réalisation et l'évaluation des interfaces homme-machine ont été rassemblés dans la dernière partie.

## **1.1. POSITIONNEMENT DE LA RECHERCHE SUR LE PLAN MÉTHODOLOGIQUE**

Un des soucis actuels des industriels est de réduire les coûts des développements logiciels et cela par la pratique du génie logiciel. A ce sujet, l'Association Française pour le Contrôle Industriel de la Qualité (AFCIQ) définit le génie logiciel comme "l'ensemble des procédures, méthodes, langages, ateliers, imposés ou préconisés par les normes adaptées à l'environnement d'utilisation, afin de favoriser la production et la maintenance de composants logiciels de qualité".

Le génie logiciel offre plusieurs modèles de développement de logiciels. Dans un premier temps, ces modèles seront parcourus afin d'étudier dans quelle mesure les facteurs humains inhérents aux interfaces y sont considérés. Au vu des lacunes de ces modèles, un cadre méthodologique sera choisi, autour duquel les parties suivantes de ce chapitre seront plus facilement positionnées.

### **1.1.1. Etude critique des principaux modèles et méthodes de développement issus du génie logiciel**

Les trois modèles les plus utilisés actuellement pour le développement de logiciels sont le modèle cascade, le modèle en "V" et le modèle spirale. Ils ne seront que brièvement décrits dans la mesure où ils ont été largement détaillés dans de nombreux ouvrages de génie logiciel. Après une étude critique de ces modèles, il sera possible de les positionner dans différentes méthodes dites de conception ou de développement logiciel, afin de discuter du degré et du moment d'implication des utilisateurs dans ces méthodes.

#### ***1.1.1.1. Le modèle cascade***

Le modèle cascade (BOEHM, 1981) est un des premiers modèles apparus pour répondre aux besoins industriels en termes de productivité et de qualité logicielle.

Certainement le plus utilisé dans les entreprises actuellement, il décrit le cycle de vie du logiciel comme une succession de huit étapes qui peuvent être grossièrement résumées de la manière suivante, figure 1.1 : (1) faisant suite à une analyse des besoins, l'étude de faisabilité débouche sur un cahier des

charges, (2) la spécification fonctionnelle du logiciel définit ce qui doit être fait, (3) la conception du logiciel, appelée également conception préliminaire, explique comment les fonctions seront décomposées pour satisfaire la spécification, (4) la conception détaillée précise les fonctions précédemment décrites, (5) le codage consiste à traduire à l'aide d'un langage chaque fonction, (6) l'intégration de chaque fonction dans le logiciel est ensuite effectuée, puis (7) le logiciel est mis en œuvre, (8) il est alors possible d'exploiter le logiciel et de le maintenir.

Chaque étape du modèle est reliée soit à l'étape suivante pour représenter l'enchaînement et la progression dans le développement, soit à l'étape précédente pour montrer les corrections pouvant survenir suite à la détection d'erreurs. Il impose donc plusieurs contraintes (NASSIET, 1987). En effet, une nouvelle étape ne commence que lorsque la précédente est achevée, ce qui nécessite des étapes de validation ou de vérification pour passer d'une étape à l'autre. De plus, les retours arrière se limitent à un retour sur la phase immédiatement antérieure. On s'aperçoit que le contrôle imposé à la fin de chaque étape ne favorise pas les changements, et tend à stabiliser rapidement le produit.

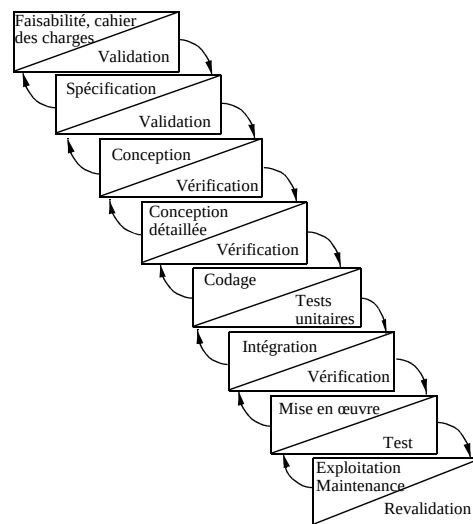


Figure 1.1 : Modèle cascade de développement logiciel

Vis-à-vis du développement d'interfaces homme-machine, une critique fondamentale peut être formulée : proposé en fin des années 70, à une époque où la prise en compte des interactions homme-machine était reléguée au second plan, il n'est pas étonnant de constater que ces aspects ne ressortent pas du logiciel global, même si celui-ci comporte une forte composante interactive.

Ainsi, à part dans la première étape où l'analyse des besoins doit logiquement concerner les utilisateurs, ceux-ci ne sont plus véritablement considérés dans les étapes suivantes qui s'adressent plutôt aux concepteurs. On retrouve tout de même les utilisateurs dans les étapes finales, et l'évaluation du produit réalisé n'est possible que lorsque celui-ci est livré à ses utilisateurs. De plus, alors que le logiciel peut être réalisé pour être utilisé pour des tâches très complexes, aucune analyse et modélisation des tâches n'est préconisée. Rejoignant la remarque précédente, aucune analyse et modélisation des utilisateurs potentiels n'est citée. C'est en fait selon le bon sens des concepteurs les plus expérimentés et de manière très informelle que ces notions extrêmement importantes sont considérées, lors de la première étape. Notons également que, dans les systèmes complexes où il est très difficile, voire impossible, de recenser exhaustivement les situations normales et anormales dans lesquelles le logiciel sera utilisé, l'impossibilité d'effectuer des retours en arrière autres que sur l'étape immédiatement précédente constitue un frein à l'efficacité du système final.

Au vu de ces critiques de fond, il va sans dire que le modèle cascade a besoin d'être revu et largement précisé afin de s'adapter à la problématique du développement de logiciels où l'interface homme-machine joue un rôle important.

### 1.1.1.2. Le modèle en V

L'Association Française pour le Contrôle Industriel de la Qualité (AFCIQ) situe le développement d'un logiciel dans celui plus général d'un système. La démarche prend la forme d'un V, inspiré de propositions datant du début des années 80 (GOLDBERG et ROBSON, 1983).

Le cycle est découpé en six phases appelées, figure 1.2 : (1) orientation, faisabilité des besoins, (2) conception du système et validation des besoins, (3) développement logiciel du système, (4) intégration du matériel et du logiciel, production du système, (5) recette et validation du système et (6) maintenance du système. Sur la figure 1.2, on peut remarquer que les phases (2) et (3) sont elles-mêmes décomposées en sous-phases plus détaillées.

Les étapes présentées pour le modèle cascade restent globalement valables pour le modèle en V. Cependant, comme le précise CALVEZ (1990), la spécification et la conception s'intègrent dans une démarche *descendante*, alors que les validations et tests seront situés dans une démarche qualifiée *d'ascendante*. Ainsi, dans chaque étape de la phase descendante doivent être prévus le plan, les moyens et les méthodes permettant d'évaluer et de valider l'étape. Ce souci de prévoir le plus en amont possible l'évaluation du système, et ceci précisément vis-à-vis de chaque phase, constitue un point fort indéniable du modèle en V. Un autre avantage consiste à considérer le développement du logiciel comme partie intégrante du développement d'un système complet, ce qui est plus réaliste que le modèle précédent, par rapport à de nombreuses applications industrielles conciliant développements logiciels et matériels.

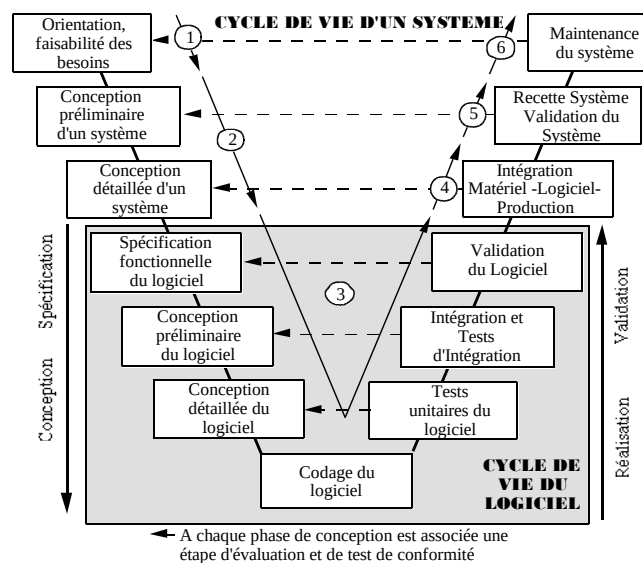


Figure 1.2 : Modèle en V du cycle de développement d'un système

Malgré ces points forts, et comme pour le modèle cascade, des critiques subsistent lorsque le point de mire du développement constitue un logiciel à forte composante interactive : l'analyse et la modélisation des utilisateurs et des tâches humaines ne sont pas positionnées. Cependant, en raison de l'importance accordée à l'évaluation dans ce modèle, l'utilisateur peut et doit jouer un rôle plus déterminant que dans le modèle cascade. En conclusion, ce modèle doit également être affiné si le développement vise des logiciels fortement interactifs.

### 1.1.1.3. Le modèle spirale

Le modèle spirale de développement du logiciel a été introduit par BOEHM et al. (1984), figure 1.3. Contrairement aux deux premiers modèles, il est axé sur *l'analyse du risque*, alors que les modèles

précédents sont guidés par la rédaction de documents dont la validation constituent le critère déterminant pour passer d'une étape à une autre.

Le modèle spirale reprend les points suivants : objectifs, contraintes, alternatives, risques, méthodes de résolution du risque, résultats de la résolution du risque, plan pour la phase suivante, engagement (humain, matériel, logiciel, financier...). Le développement est décrit comme un processus itératif selon quatre phases (une par cadran) illustré par le parcours depuis l'intérieur, jusqu'à l'extérieur de la spirale, figure 1.3 : (i) les objectifs, les alternatives et les contraintes sont d'abord déterminés, (ii) les alternatives relatives aux objectifs et aux contraintes sont évalués, à l'aide de prototypes, simulations, essais ou toute autre méthode permettant d'évaluer le risque engendré par chaque alternative, (iii) suite à l'étape précédente, deux cas de figures se présentent : soit il reste des risques et dans ce cas il faut poursuivre la spirale (ce qui fait l'objet de la phase suivante), soit on aboutit à un prototype opérationnel qui servira de base aux étapes traditionnelles présentées lors de la description du modèle cascade, (iv) les moyens à mettre en œuvre pour poursuivre la spirale sont fixés. De plus, le modèle spirale comporte une composante radiale représentant les coûts cumulés pour mener à bien les différentes étapes ainsi qu'une composante angulaire indiquant les progrès réalisés à la fin de chaque étape du projet.

D'après POULAIN (1994), ce modèle possède un avantage certain vis-à-vis du développement de logiciels fortement interactifs : il favorise l'expression de spécifications qui ne sont pas nécessairement aussi exhaustives que pour les modèles précédents. En effet les besoins sont formulés progressivement, et les différents risques rencontrés sont résolus au fur et à mesure. Ce n'est qu'une fois les risques stabilisés que l'on procède à une étape plus fine, ce qui a pour effet de retarder l'élaboration détaillée d'éléments logiciels de moindre risque tant que les éléments à hauts risques n'ont pas été résolus. C'est pourquoi de nombreux auteurs concernés par le développement d'applications interactives, tels CROCHARD et WALFARD (1991), VANNESTE et LAMBERT (1991), JAMES (1991), MEINADIER (1991) ou NIELSEN (1993) s'accordent à penser que le prototypage, à défaut d'une méthode de conception-évaluation universelle, systématique et rigoureuse, propose une solution très pratique pour éviter ou tout au moins limiter les défauts de conception et les erreurs ergonomiques, pour accroître la qualité des produits et abaisser les coûts de production, de mise au point et de maintenance.

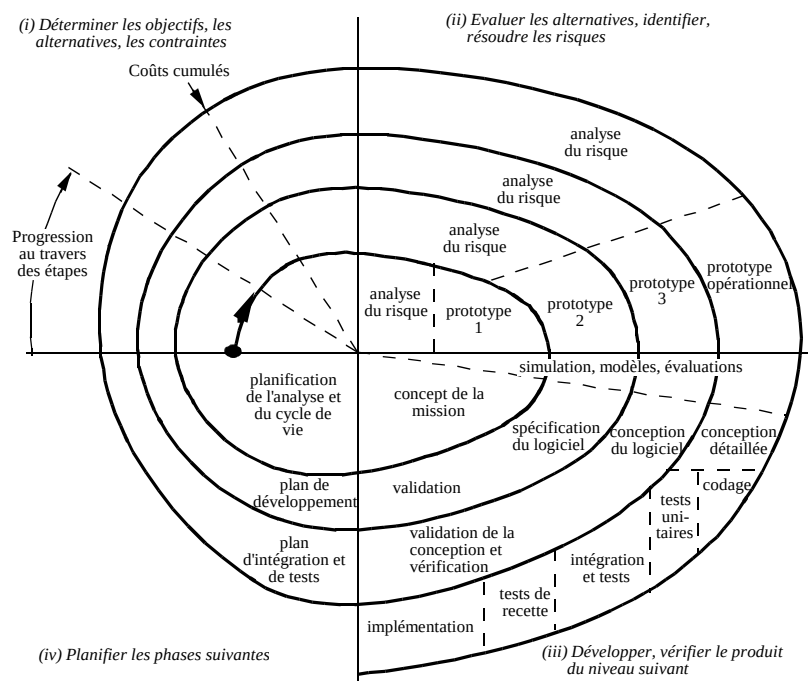


Figure 1.3 : Modèle spirale de développement de logiciel

Ainsi, dans la mesure où le modèle spirale introduit dès le début du cycle de vie la possibilité d'évaluer à partir d'un prototype les choix réalisés, il laisse entrevoir des perspectives prometteuses pour le développement des interfaces homme-machine dans les systèmes complexes. Toutefois, ce modèle est tout de même critiquable : comme pour les modèles précédents, les interfaces ne sont pas citées même si la démarche même les sous-entend, et l'analyse et la modélisation des utilisateurs et des tâches humaines ne sont pas préconisées et sont laissées à l'appréciation des concepteurs.

Ces trois modèles de développement sont utilisés tels quels lors de nombreux projets. Ils sont aussi et surtout à la base d'un ensemble de méthodes dites de conception ou de développement, dans lesquelles il est maintenant utile de préciser le degré et le moment d'implication de l'utilisateur.

#### **1.1.1.4. Etude du degré et du moment d'implication de l'utilisateur dans les méthodes de conception**

Dans un article fort intéressant, PASCOT et BERNADAS (1993) ont comparé 6 méthodes (ou ensemble de méthodes) de conception de systèmes d'information informatisés selon quatre dimensions elles-mêmes décomposées en sous-dimensions : représentation, organisation, gestion de projet et technologie.

Les méthodes étudiées ont été sélectionnées en raison de leur importance, leur diffusion et la qualité et la richesse de leur documentation. On y distingue : (i) des méthodes représentatives de celles basées sur l'analyse et la conception structurée, telles SASD (YOURDON et CONSTANTINE, 1979) et JSD (JACKSON, 1983), méthodes largement répandues dans le monde, (ii) MERISE (TARDIEU et al., 1985, 1991) largement utilisées en France principalement et de plus en plus au niveau mondial, (iii) SSADM, méthode très répandue en Angleterre et dans certains pays sous influence anglaise, et qui correspond aux méthodes dites structurées, mais avec une forte préoccupation de gestion de projet (LONGWORTH et NICHOLLS, 1986), (iv) ID (Information Engineering) issue de MARTIN (1989 ; 1990), surtout connue en Amérique du Nord, influencée par les méthodes dites structurées, (v) les méthodes orientées objets en général, pour lesquelles les auteurs de l'article soulignent le manque d'intérêt accordé, la plupart du temps, à la gestion de projet (vi) et enfin la méthode DATARUN qui n'offre un intérêt que pour les auteurs de l'article apparemment.

Dans leur étude comparative, et particulièrement dans la dimension gestion de projet, *le degré d'implication des utilisateurs* correspond à un classement hiérarchique allant de "pas d'implication" à "essentiel" exprimant la façon dont on perçoit le rôle des utilisateurs dans la méthode. Le moment d'implication des utilisateurs est défini par rapport au développement (au début, au milieu ou à la fin).

Nous avons décidé d'étendre l'étude pour ces deux dimensions en ajoutant la méthode AXIAL, provenant d'IBM et visant le développement de systèmes d'information, dont des améliorations récentes visent une meilleure implication des utilisateurs (GROUPE CIC, 1989), ainsi que la méthode REMORA, méthode française de conception de systèmes d'information, dont le souci de description de l'existant est important (ROLLAND et al., 1988). De plus, au lieu de s'intéresser aux méthodes orientées objets en général, nous avons décidé de nous pencher sur trois méthodes particulières : OOA/OOD (COAD et YOURDON, 1990 ; BOOCH, 1994), OMT (RUMBAUGH et al., 1991), ces méthodes étant fortement centrées sur les aspects de modélisation au détriment de l'utilisateur, et SYS\_P\_O (JAULENT et LARRIEUX, 1993), méthode française particulièrement intéressante par son souci de veiller à la productivité du projet et à l'adéquation aux besoins exprimés. Notons que nous avons supprimé de notre étude comparative succincte la méthode DATARUN sur laquelle la documentation est inexistante à notre connaissance. Le résultat de cette nouvelle comparaison est visible dans le tableau 1.1. Pour chaque méthode, le modèle de développement supportant celle-ci est indiqué, en plus d'une évaluation du degré et du moment d'implication de l'utilisateur.

Plusieurs remarques découlent de ce tableau. D'abord, le modèle spirale n'est utilisé que dans les méthodes orientées objets, et on ne trouve pas trace du modèle en V. Ensuite, durant de nombreuses années, l'utilisateur se contentait d'intervenir lors de la livraison du produit et/ou lors de l'analyse des besoins. On s'aperçoit que, à part IE et SYS\_P\_O, aucune méthode actuelle ne le fait vraiment intervenir du début à la fin du projet. Il reste de manière générale peu impliqué, ce qui rejoint le constat déjà ancien, mais toujours d'actualité, de BARTHET (1988). Toutefois, on assiste actuellement à un frémissement concernant les facteurs humains en génie logiciel, qui se confirme dans les ouvrages récents, tel celui de SOMMERVILLE (1994), et qu'on remarque au niveau de certaines méthodes, telles MERISE et AXIAL.

Tableau 1.1. Etude comparative de neuf méthodes de développement

METHODE	Cycle de développement supportant la méthode	Degré d'implication de l'utilisateur	Moment d'implication de l'utilisateur
SASD , JSD	Cascade	Peu	Début
MERISE	Cascade	Peu à beaucoup	Début, fin (un peu)
SSADM	Cascade	Peu	Début, milieu
IE	Cascade	Beaucoup	Début, milieu, fin
AXIAL	Cascade	Peu à beaucoup	Début, fin
REMORA	Cascade	Peu	Début, fin (un peu)
OOA/OOD	Spirale	Peu	Fin
OMT	Spirale	Peu	Fin
SYS_P_O	Spirale	Beaucoup	Début, milieu, fin

#### 1.1.1.4. Conclusion sur les modèles et méthodes issus du génie logiciel

Les trois modèles principaux de développement issus du génie logiciel, en l'occurrence le modèle cascade, le modèle en V et le modèle spirale, offrent un cadre général très utile pour les concepteurs. Ce n'est pas un hasard d'ailleurs si ces modèles ont été pris en compte lors de dizaines de milliers de projets informatiques depuis le début des années 80. Ces modèles sont à la base de nombreuses méthodes visant à faciliter la couverture de l'ensemble des étapes du cycle de vie du logiciel, en offrant un cadre de travail rigoureux et systématique.

Cependant, si on s'intéresse de plus près à ces méthodes et modèles, et ceci vis-à-vis du développement d'applications fortement interactives, des lacunes apparaissent. Ceci est particulièrement vrai lorsque les applications interactives sont destinées à être utilisées dans des systèmes industriels complexes où les contraintes sont considérables (milliers de variables à surveiller, tâches humaines fortement cognitives, nombreuses situations normales et surtout anormales à considérer, critères de sécurité et de production à respecter, etc). D'ailleurs, il est bien connu que les méthodes recensées sont pour la plupart plus adaptées au développement de systèmes d'information qu'à celui des applications interactives qui nous préoccupent.

C'est pourquoi il est maintenant nécessaire de définir un premier cadre méthodologique dans lequel les aspects liés aux interfaces homme-machine sont plus explicitement considérés et pas seulement sous-entendus comme dans certains des modèles ou méthodes précédents.

### 1.1.2. Cadre théorique et méthodologique pour le développement des interfaces homme-machine

Des recherches menées depuis plusieurs années au LAMIH visent à concevoir et évaluer des interfaces homme-machine permettant à leurs utilisateurs de superviser un système industriel complexe, tout en tenant compte de critères de performance, de sécurité et de confort imposés au système homme-machine. Ces travaux s'appuient sur une démarche originale, appelée méthode en U, constituée de deux phases (MILLOT, 1990 ; ABED et ANGUE, 1990 ; MILLOT et DEBERNARD, à paraître) : (i) une phase descendante de modélisation le système homme-machine, et qui aboutit à sa mise en œuvre, (ii) une phase ascendante consistant en l'évaluation du système global.

#### 1.1.2.1. Méthode en U, phase descendante de conception et réalisation

Dans la première phase résumée en figure 1.4, la démarche consiste dans un premier temps à analyser le système à interfacier. Dans ce but, il existe différentes méthodes permettant de mieux appréhender le fonctionnement de celui-ci et des sous-systèmes qui le composent. On distingue généralement deux grands types de méthodes (FADIER, 1990) :

- Les premières, généralement bien connues des informaticiens, sont destinées à l'analyse du système en fonctionnement normal et à sa description selon des aspects structurels et fonctionnels. Citons par exemple SADT (IGL TECHNOLOGY, 1989), SA-RT (HATLEY et PIRBHAI, 1991) ou SA (DeMARCO, 1979).
- D'autres méthodes, provenant surtout des domaines de la maintenance et de la fiabilité des systèmes, sont utilisées pour analyser le système en fonctionnement anormal. Les plus connues sont l'AMDEC (Analyse des Modes de Défaillances, de leurs Effets et de leur Criticité) inspirée de RECHT (1966), et les arbres de défaillances (HASSL, 1965). L'utilisation de telles méthodes vise à définir les différents cas prévisibles de dysfonctionnements et donc de préparer l'étape suivante d'analyse des tâches humaines.<sup>4</sup>

Une fois la modélisation du système effectuée, une analyse des tâches humaines prescrites a pour objectif d'établir les activités que les utilisateurs auront à effectuer. Celle-ci doit tenir compte du modèle des différents utilisateurs, en termes de limites et de ressources physiques et cognitives, relatives à l'acquisition et au traitement de l'information, mais également à l'exécution des actions.

Cette analyse des tâches devra mettre en évidence les besoins informationnels des utilisateurs, correspondant aux données nécessaires pour effectuer les différentes tâches, ainsi que leurs besoins en outils (fonctions) d'assistance, pouvant prendre la forme d'outils d'aide à la décision (filtrage d'alarmes, diagnostic, planification, etc). L'intégration de tels outils induit dans la majeure partie des cas des problèmes de coopération entre ces outils d'assistance et les utilisateurs, cette coopération s'effectuant au travers des interfaces homme-machine.

---

<sup>4</sup> A partir d'un recensement des défaillances susceptibles d'affecter un système, l'AMDEC aide à évaluer les effets de chaque mode de défaillance des composants du système sur les différentes fonctions de celui-ci, et d'identifier les modes de défaillance ayant des conséquences sur la disponibilité, la production, la fiabilité, la maintenabilité ou la sécurité du système. Sont ainsi mis en évidence pour chaque mode de défaillance, les causes, les effets, les moyens de compensation existants et la criticité. La méthode de l'Arbre des Causes est une méthode déductive, encore appelée méthode de l'Arbre de Défaillances, des Défauts ou des Fautes. Cf. l'ouvrage de VILLEMEUR (1988) qui recense l'ensemble de ces méthodes d'analyse et donne de nombreux exemples.



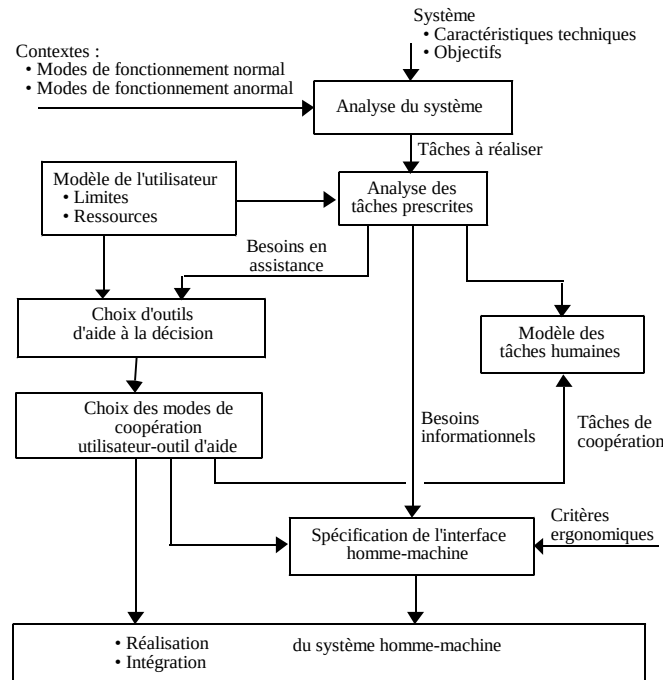


Figure 1.4 : Phase descendante (adaptée de MILLOT et ROUSSILLON, 1991)

A partir de la mise en évidence des besoins informationnels et besoins en assistance, il est possible de définir une architecture pour l'interface homme-machine et de spécifier celle-ci : nombre de vues, enchaînements de celles-ci, rôle (supervision, diagnostic, action, démarrage...) et contenu informationnel de chaque vue, modalité des dialogues homme-machine, etc. La spécification doit tenir compte d'un ensemble de critères issus de l'ergonomie des logiciels, relatifs par exemple au codage de l'information, à la cohérence, à la lisibilité, aux différents modes de représentation possibles, etc, en visant à éviter au maximum les sources d'erreurs humaines provenant par exemple de problèmes de perception, d'identification ou d'utilisation d'informations.

La spécification des interfaces homme-machine et des différents outils d'aide conduit à la réalisation et l'intégration du système homme-machine complet ou d'un *prototype* de celui-ci sur le site et/ou en simulation.<sup>5</sup>

### 1.1.2.2. Méthode en U, phase ascendante d'évaluation

La phase d'évaluation (figure 1.5) vise à terme la validation du système interactif conçu et réalisé lors de la phase descendante. D'après MILLOT et ROUSSILLON (1991), les critères d'évaluation tiennent compte des performances du système global, selon d'une part les aspects liés à *l'utilisabilité*<sup>6</sup> du système homme-machine, et d'autre part les performances du système en terme d'écarts entre la production et les objectifs.

Cette évaluation impose la définition de protocoles expérimentaux rigoureux, qui ont pour but de définir la manière dont les tests seront conduits, mais aussi les données qui seront collectées (PHILIPPE,

<sup>5</sup> Le lecteur remarquera que dans cette méthode, on ne retrouve pas explicitement les étapes classiques du génie logiciel concernant le développement du système et/ou du logiciel (conception préliminaire, conception détaillée, codage, tests unitaires, tests d'intégration, etc). Celles-ci seront intégrées par la suite dans les boîtes appelées "Spécification", et "Réalisation/intégration", comme nous le détaillerons aux chapitres suivants.

<sup>6</sup> Voir à ce sujet GOULD (1988) et RAVDEN et JOHNSON (1989) qui listent des critères d'utilisabilité d'un système interactif. Alors que RAVDEN et JOHNSON se focalisent sur des critères essentiellement liés aux aspects interactifs, GOULD va plus loin en ajoutant la documentation, l'installation du logiciel ou encore la maintenance. Notons par ailleurs un intéressant ouvrage de J. NIELSEN, paru en 1993 sur l'utilisabilité et intitulé "Usability Engineering".

1993). D'après MILLOT (1990), cette étape n'est pas simple dans la mesure où certaines données ne sont pas directement observables et ne permettent pas de mesurer les difficultés rencontrées par les utilisateurs. L'analyse cognitive des activités et le traitement des données qui en résulte peuvent déboucher sur une comparaison entre les tâches effectuées réellement par les utilisateurs et les tâches prescrites définies précédemment (ABED, 1990). Cela conduit soit à valider le système homme-machine soit à mettre en évidence ses carences et à affiner progressivement celui-ci, particulièrement au niveau des interfaces homme-machine et des outils d'aide.

En raison de nombre élevé de méthodes permettant d'évaluer un logiciel interactif, celles-ci feront l'objet d'une partie spécifique, cf. 1.5. Certaines sont déjà bien connues des informaticiens, telles que les maquettes, les tests papiers, les questionnaires d'utilisabilité, le prototypage, les interviews, les cahiers de doléances (MEINADIER, 1991). De nombreuses autres, par contre, sont moins connues mais trouveront avantageusement leur place dans les méthodes du génie logiciel. Cette remarque sera discutée plus en profondeur plus loin dans ce chapitre.

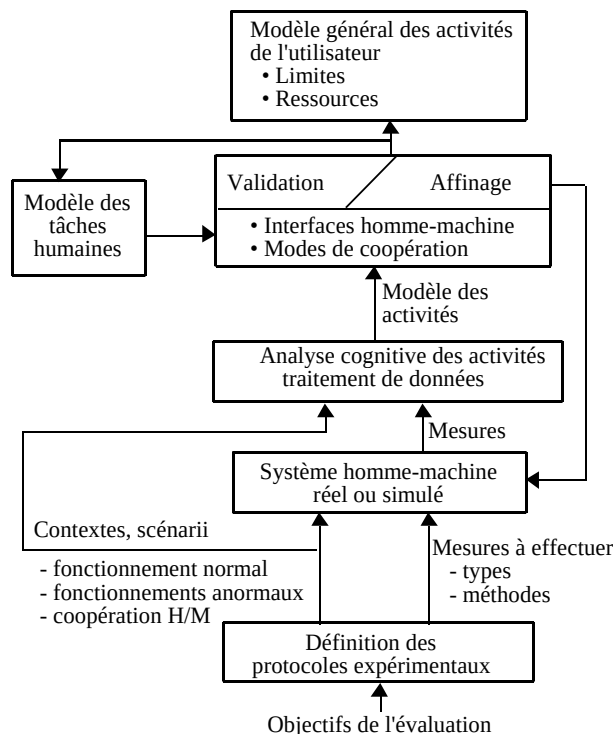


Figure 1.5 : Phase ascendante d'évaluation (MILLOT et ROUSSILLON, 1991)

### 1.1.2.3. Conclusion sur le cadre méthodologique choisi

De par les notions relatives aux interfaces homme-machine traitées considérées explicitement dans la méthode en U, le choix de cette méthode, comme cadre théorique et méthodologique pour ce mémoire s'impose logiquement. Cette méthode générale permet de mieux positionner tout un ensemble de notions essentielles qui seront plus détaillées dans la suite de ce chapitre : (i) l'analyse et la modélisation de l'utilisateur et des tâches humaines dans la partie 1.2, (ii) les différentes architectures possibles pour l'application interactive dans la partie 1.3, (iii) la spécification des interfaces homme-machine, les outils de réalisation et les méthodes d'évaluation dans la partie 1.4.

Cependant, dans la mesure où elle ne détaille pas les aspects liés au développement, selon les étapes propres au génie logiciel, il sera intéressant dans le dernier chapitre de l'affiner, puis d'unifier la méthode en U avec les modèles issus du génie logiciel.

## 1.2. ANALYSE ET MODÉLISATION DES TÂCHES HUMAINES ET DE L'UTILISATEUR

L'analyse et la modélisation des tâches humaines d'une part, et de l'utilisateur d'autre part, sont étroitement liées, les limites et ressources cognitives et physiques des utilisateurs contribuant directement à leur efficacité et à leur fiabilité vis-à-vis des tâches à accomplir. Dans un but pédagogique, nous avons tout de même séparé ces notions en deux sous-parties successivement commentées. Toutefois, le passage d'une notion vers l'autre sera souvent sous-jacente.

### 1.2.1. Analyse et modélisation des tâches humaines

L'importance accordée à l'analyse et la modélisation des tâches humaines est grandissante, conduisant à des recherches actives depuis plusieurs années en informatique et dans les sciences cognitives, et ceci dans un but de développement d'applications interactives (voir par exemple DIAPER, 1989 ; CARTER, 1991 ou MCGREW, 1991). La figure 1.6 rappelle trois cas-types où l'analyse de tâche trouve particulièrement son intérêt :

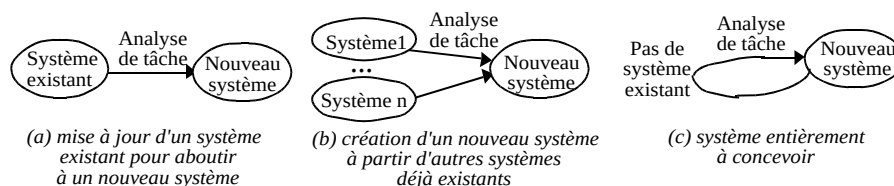


Figure 1.6 : Cas-types d'analyse de tâche (STAMMERS et al., 1990)

- (a) L'analyse de tâche pour la mise à jour d'un système existant est toujours très fructueuse. Dans le cas où les utilisateurs du système sont présents, ils peuvent apporter leur connaissance et leur expérience sur celui-ci.
- (b) Lorsque le système à concevoir doit résulter de plusieurs systèmes déjà existants, il est souvent avantageux d'analyser ceux-ci dans le but de déduire des informations réutilisables pour le nouveau système.
- (c) Lorsque le nouveau système est entièrement à concevoir, et qu'il n'existe pas d'autre système de référence, l'identification des tâches est essentielle. Cependant, il s'agit de voir dans cette démarche un processus de conception de tâches plutôt que de description.

Il s'agit d'abord de positionner l'analyse et la modélisation des tâches humaines en tant que travail d'équipe. Puis, l'analyse telle qu'elle est préconisée en génie logiciel sera discutée. Il sera possible ensuite de parcourir plusieurs méthodes de modélisation des tâches.

#### 1.2.1.1. Analyse et modélisation en tant que travail d'équipe

Au niveau de l'analyse de tâches humaines, une collaboration étroite et pluridisciplinaire est particulièrement nécessaire entre différents intervenants du processus de développement de l'application interactive (NORMAN, 1986 ; MEINADIER, 1991 ; POULAIN, 1994). Dans les systèmes industriels complexes en particulier, quatre types de connaissances s'avèrent particulièrement complémentaires :

- *Les connaissances de l'utilisateur* : elles se basent sur la pratique et l'expérience de différentes situations de supervision du système complexe. L'utilisateur connaît l'efficacité de certaines interventions, il a déjà dans le meilleur des cas utilisé différents outils informatiques d'assistance. Il

est donc théoriquement en mesure de recenser une grande partie des tâches qui lui ont été assignées et d'en décrire les contraintes.

- *Les connaissances de l'ingénieur* : il connaît les installations, leur fonctionnement, leur comportement, leurs composants et les influences entre variables. Il est capable de recenser certaines informations utiles à l'utilisateur et de contribuer à la définition des modes opératoires via l'interface, et des tâches humaines qui en découlent.
- *Les connaissances du spécialiste des facteurs humains* : il est familier des exigences physiques et cognitives, ainsi que des limites et des ressources théoriques d'un utilisateur réalisant une tâche donnée dans un environnement socio-technique. Il est donc d'un apport précieux lors de l'analyse et de la modélisation des tâches humaines.
- *Les connaissances de l'informaticien* : il apporte ses connaissances liées aux méthodes et outils de développement. Il doit en outre faciliter la liaison entre les données recueillies et leur modélisation, à l'aide de méthodes du type de celles expliquées en 1.2.1.3.

En partant de cette considération, et comme il a été suggéré lors de la présentation de la méthode en U (cf. 1.1.2), le fait de construire un modèle des tâches prescrites et un modèle des activités possède l'avantage de faciliter la comparaison de ces deux modèles. Il est alors possible de mettre le doigt sur des dysfonctionnements, des oublis, des sources d'erreur, et donc d'en déduire un ensemble de spécifications visant en particulier l'aménagement des interfaces homme-machine et des outils d'assistance (MILLOT, 1990 ; ABED, 1990). Notons que dans le cas d'un fonctionnement anormal d'un système complexe, l'analyse des tâches humaines (suivie d'une modélisation rigoureuse) peut faciliter la détermination des parades à mettre en place pour compenser ce fonctionnement anormal, ces parades pouvant être confiées à l'utilisateur au travers de l'interface homme-machine. Il devient clair que l'analyse et la modélisation des tâches humaines doivent jouer un rôle essentiel dans le processus de développement, ce qui est loin d'être pratiqué aujourd'hui.

#### ***1.2.1.2. Analyse des tâches vue des méthodes et des modèles du génie logiciel***

Dans un tel cadre pluridisciplinaire, l'analyse peut s'effectuer à l'aide de techniques variées : analyse de documents, questionnaires, observations et analyses de protocoles sur site de situations de travail, complétées par des verbalisations durant des entretiens avec les différents intervenants du processus de développement... Ainsi, l'analyse consiste à recenser des données subjectives et objectives relatives aux différentes tâches, à leur déroulement, à leurs contraintes, aux besoins des utilisateurs pour les réaliser, etc.

Concernant une analyse de tâches, de nombreuses questions se posent pour l'analyste informaticien : "Comment procéder ? Quelles questions poser ? Par qui faut-il commencer ? Qui et quoi faut-il observer, pendant combien de temps, lors de quelles situations ? Existe-t-il différents types d'utilisateurs ? Comment analyser les données objectives et subjectives résultant de l'analyse ? Quels sont les documents qu'il faut consulter, et pourquoi ? A quoi faut-il faire attention ?..."

Malgré un nombre grandissant d'ouvrages consacrés aux méthodes de développement, l'analyste trouve peu de réponses précises à ses questions. Les explications fournies pour cette phase cruciale sont souvent vagues et chargées de sous-entendus, comme si chaque lecteur était un spécialiste de la question. *Il faut commencer par une analyse des besoins des utilisateurs*. D'accord. Mais comment ? La situation est la suivante :

- Nous avons vu dans la partie 1.1 que l'analyse et la modélisation des tâches humaines n'étaient pas représentées explicitement dans les principaux modèles de développement logiciel (cascade, en V, spirale).

- A notre connaissance, pratiquement aucune directive ou démarche précise n'est fournie avec les méthodes d'analyse (et/ou de spécification) systémiques, dont les plus connues sont certainement SADT, SA ou SA-RT. Pourtant, comme nous en verrons un exemple ci-après avec SADT (cf. 1.2.1.3.1), elles peuvent être d'un apport important.
- En ce qui concerne les méthodes orientées objet couvrant au minimum l'étape d'analyse, telles OOA (couvrant l'étape d'analyse) ou OMT (couvrant l'analyse, la conception et l'implémentation), pratiquement aucun conseil n'est donné non plus concernant l'analyse et la modélisation des tâches humaines.
- C'est plutôt dans des ouvrages sur certaines méthodes visant à couvrir l'ensemble des étapes du cycle de vie du système ou du logiciel, qu'on trouve trace de l'analyse et de la modélisation de tâches humaines. Par exemple, dans la phase d'analyse de l'existant de la méthode MERISE, des lignes directrices sont fournies pour l'analyse des tâches, et un symbolisme est proposé pour leur modélisation. Le lecteur se référera à ce sujet à l'ouvrage de COLLONGUES et al. (1986). Notons également un chapitre fort intéressant intitulé "méthodologie de l'étude de l'existant" dans l'ouvrage collectif du groupe GALACSI (Groupe d'Animation et de Liaisons pour l'Analyse et la Conception de Systèmes d'Information) en 1986. Il existe bien sûr d'autres exemples, mais en nombre insuffisant.

Malgré de nombreuses recherches prometteuses menées dans le domaine de l'analyse et de la modélisation des tâches, on s'aperçoit donc que peu de recommandations et de lignes directrices précises se retrouvent dans les méthodes les plus connues du génie logiciel. La démarche de l'informaticien analyste, lors de la phase si cruciale "d'analyse des besoins" ou "d'analyse de l'existant" reste donc très informelle, et s'avère souvent source d'incomplétude. Son expérience et sa subjectivité influenceront alors largement et inévitablement sur la manière et l'efficacité dont les besoins des utilisateurs seront analysés et spécifiés pour les différentes tâches à accomplir. Pourtant, des méthodes rigoureuses d'analyse de tâche existent.

Dans les sciences cognitives, et particulièrement en ergonomie et en psychologie du travail, de telles méthodes sont maîtrisées depuis longtemps. La notion d'analyse de la tâche, ainsi que des méthodes opérationnelles ou des points de repère, sont expliquées dans de nombreux ouvrages, tels ceux de FAVERGE (1972), GRANDJEAN (1983), PINSKY et THEUREAU (1985), WILSON et CORLETT (1990), THEUREAU (1991), AMALBERTI et al. (1991) ou SPERANDIO (1988 ; 1991). DANIELLOU (1986), quant à lui, donne des orientations et des recommandations pour l'analyse de *l'activité future probable* des utilisateurs.

Il faut noter également que certaines données peuvent être recueillies à l'aide des techniques d'acquisition de connaissances usitées en Intelligence Artificielle. Pour la description de ces techniques, le lecteur se référera par exemple à NASSIET (1987), AUSSÉNAC (1989), DIENG (1990) ou BENKIRANE (1991). On retiendra aussi un article de synthèse de BENYSH et al. (1993) qui recense ces techniques en les mettant en rapport avec la conception d'interface.

Toutes les données recueillies lors de l'analyse des tâches pourront faire l'objet d'une modélisation, en vue d'en tirer des recommandations pour la conception.

### **1.2.1.3. Méthodes de modélisation des tâches humaines**

Plusieurs méthodes sont applicables pour modéliser les différentes tâches à effectuer par l'utilisateur, et faciliter ainsi l'analyse des besoins informationnels et la spécification des interfaces homme-machine et des outils d'aide. Deux d'entre elles, parmi les plus connues, seront données à titre d'exemple : la première

est basée sur deux méthodes bien connues (SADT et réseaux de Petri), la seconde appelée MAD propose une description des tâches selon des concepts facilement maîtrisables.

#### 1.2.1.3.1. Méthode basée sur SADT et les réseaux de Petri

Celle-ci est basée sur l'utilisation de deux méthodes bien connues des informaticiens. Elle a été développée au LAMIH, pour répondre initialement à un besoin d'analyse de tâches humaines de contrôle du trafic aérien (ABED, 1990). Cette méthode repose sur une décomposition hiérarchique descendante de plus en plus précise. A l'aide d'actigrammes SADT, il est possible de décrire l'enchaînement des tâches, les données en entrée, en sortie, celles contrôlant la tâche, ainsi que les mécanismes ou supports d'activité exprimant les moyens utilisés pour exprimer la tâche.

A l'aide de SADT, il n'est pas possible de représenter la composante dynamique des tâches, c'est pourquoi l'utilisation des réseaux de Petri s'avère indispensable et complémentaire de SADT. Rappelons qu'un réseau de Petri est un graphe constitué de transitions et de place reliés par des arcs orientés. La description sous la forme de réseau de Petri intervient lorsque le niveau de description sous la forme de SADT est suffisamment fin pour faire apparaître une tâche exécutable par l'utilisateur. Un exemple de description à l'aide d'un réseau de Petri d'une tâche terminale de supervision d'un réseau ferroviaire est visible en figure 1.7 (BENAISSA et al., 1992).

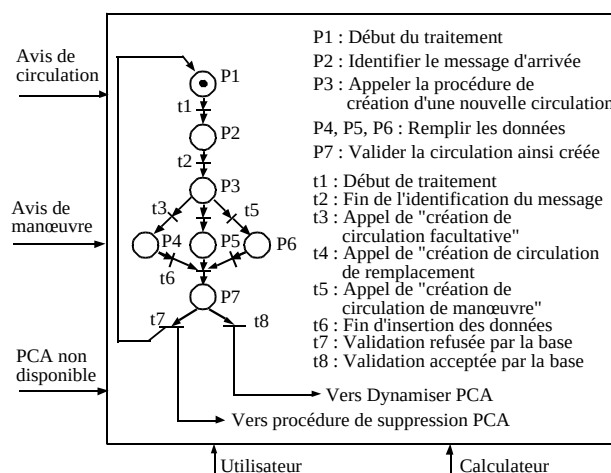


Figure 1.7 : Modélisation par SADT d'une tâche terminale

#### 1.2.1.3.2. M.A.D. : Méthode Analytique de description des tâches

Issue de travaux menés à l'INRIA par SCAPIN et PIERRET-GOLBREICH (1989), MAD vise la description de tâches humaines dans un but de meilleure prise en compte de l'ergonomie dans la conception d'interfaces utilisateurs. Les principaux concepts introduits dans le formalisme M.A.D. sont ceux de tâche, d'action et de structure. Le concept de tâche est représenté par un objet générique appelé objet-tâche et composé d'un état initial, d'un état final, d'un but, des pré-conditions et des post-conditions. Ainsi, l'objet tâche est la racine de deux sous classes, la classe "tâche-élémentaire" et la classe "tâche-composée" :

- La tâche élémentaire est une tâche indécomposable, dont le niveau opérationnel est caractérisé par un objet-méthode de type simple, c'est-à-dire une action.
- La tâche composée est une tâche dont le niveau opérationnel est défini par une structure décrivant le corps de la tâche. Le concept de structure est représenté par un objet générique caractérisé par un constructeur décrivant l'agencement des différentes tâches impliquées et les arguments du constructeur. Ainsi, plusieurs constructeurs ont été définis, tels que : SEQ : tâche séquentielle,

PAR : tâches parallèles, ALT : tâches alternatives, BOUCLE : tâches itératives et FAC : tâches facultatives.

La figure 1.8. donne le principe de description d'une tâche humaine dans le domaine du contrôle aérien (EL FAROUKI, 1991). Sur la partie gauche apparaît la décomposition sous forme d'arbre de la tâche "Gérer la position", alors que sur la partie droite apparaît sa description. Cette méthode a d'abord été testée sur des tâches de bureau pour ensuite être utilisée pour décrire des tâches plus complexes de contrôle de la navigation aérienne.

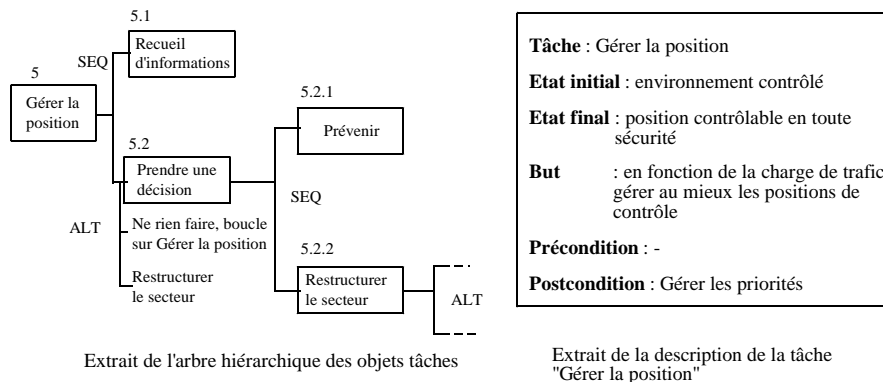


Figure 1.8. Description par MAD de tâches humaines

#### 1.2.1.3.3. Conclusion sur les méthodes de modélisation des tâches

Il aurait été possible de décrire d'autres méthodes connues. Par exemple GOMS (Goal, Operator, Method, Selection), défini par CARD et al. (1983), permet de modéliser le comportement de l'utilisateur à différents niveaux d'abstraction, depuis la tâche jusqu'aux actions physiques en termes de buts, d'actions élémentaires, de méthodes pour réaliser un but et de règles de sélection de méthodes. Citons également les travaux de SEBILLOTTE (1987) qui a utilisé le paradigme de planification hiérarchique en Intelligence Artificielle (SACERDOTI, 1977) où chaque étape individuelle dans un plan est considérée comme une tâche avec ses propres entrées, conditions, buts, etc, principes dont s'inspire la méthode MAD. Notons aussi le symbolisme proposé avec la méthode MERISE débouchant sur une description statique des tâches (COLLONGUES et al., 1986) ou encore la méthode DIANE (BARTHET, 1988), basée sur MERISE, permettant de formaliser le dialogue homme-machine à partir d'enchaînements et de déclenchements de procédures, de tâches et d'opérations. Chacune des méthodes propose une description hiérarchique, dans laquelle les sous-tâches peuvent être individuellement décrites. A notre avis, plusieurs autres méthodes bien connues en génie logiciel pourraient être également utilisées, par exemple SA ou SA-RT.

#### 1.2.1.4. Conclusion sur l'analyse et la modélisation des tâches humaines

Il est d'ores et déjà acquis que l'analyse et la modélisation des tâches constituent une étape essentielle dans le développement de systèmes interactifs dans les systèmes complexes, en facilitant par la suite l'analyse des besoins et la génération de spécifications d'interface homme-machine et de modules d'aide adaptés. C'est par une démarche pluridisciplinaire que cette étape trouvera toute son efficacité. Le lecteur a pu s'apercevoir que cette étape peut être facilitée par un ensemble de méthodes ou de principes dont certains sont maîtrisés depuis longtemps en génie logiciel. Dans la démarche d'analyse des tâches, l'analyse de l'utilisateur a été positionnée. Il est maintenant intéressant de plus la détailler, en se basant sur des tentatives de modélisation existantes.

### 1.2.2. Analyse et modélisation cognitive de l'utilisateur

Afin d'assister efficacement l'utilisateur et améliorer ainsi l'efficacité et la fiabilité du système homme-machine, il s'agit d'analyser rigoureusement puis de modéliser ses caractéristiques, sa manière de travailler, de raisonner, d'agir. Ce travail est réservé aux spécialistes des facteurs humains, et doit déboucher sur une banque de données qui sera considérée lors de la spécification et de l'évaluation du système interactif, celui-ci devant être adapté au modèle de l'utilisateur.

Ainsi doit d'abord être effectuée une classification par fonction (ou rôle) des différents intervenants concernées directement ou non par le système d'assistance. Chacun de ces intervenants pourra ensuite être caractérisé par son niveau de formation, ses connaissances et son expérience débouchant sur des manières d'appréhender le système complexe suivant les différentes situations de fonctionnement, normales ou anormales. Pour ces situations, leurs objectifs et critères de décision et d'intervention devront être connus.

En considérant que les fonctions du système d'assistance doivent être adaptées aux processus de raisonnement mis en oeuvre par l'utilisateur, l'identification de ces processus est nécessaire. Celle-ci peut d'ores et déjà s'appuyer sur des tentatives de modélisation, recensées dans la littérature depuis une quinzaine d'années. Certaines d'entre elles sont successivement présentées.

#### 1.2.2.1. Tentatives de modélisation de la mémoire humaine

Pour effectuer ses tâches, l'utilisateur a de multiples occasions d'utiliser sa mémoire, par exemple pour : prévoir l'influence d'un changement de consigne sur une variable par la connaissance du temps de réponse de celle-ci, mettre en oeuvre une procédure, constituée de différentes sous-tâches séquentielles et/ou parallèles, appréhender le déroulement d'un cycle de fonctionnement du système, utiliser dans le raisonnement des valeurs de variables qui viennent d'être lues, modifier la valeur d'une variable par rapport à des contraintes de production ou de sécurité, etc. Deux approches de modélisation de la mémoire humaine issues des sciences cognitives sont successivement décrites.

##### 1.2.2.1.1. Première approche de modélisation en 3 systèmes de stockage

Selon une première approche, on considère que la mémoire humaine est composée de trois systèmes de stockage de l'information : le registre de l'information sensorielle, la mémoire à court terme et la mémoire à long terme (LINDSAY et NORMAN, 1980).

*Le registre de l'information sensorielle* conserve une image assez précise et complète du monde, telle qu'elle est captée par les systèmes sensoriels. Ce registre est capable de contenir l'image pendant 200 ms environ. L'information constituant l'image est à l'état brut et ne peut pas être modifiée par des mécanismes cognitifs. Elle disparaît donc inexorablement quelle que soit la volonté de l'individu à la maintenir dans le registre. *La mémoire à court terme* permet de retenir, pendant plusieurs secondes, une interprétation immédiate des événements, constituant une synthèse et une structuration des informations contenues dans le registre de l'information sensorielle. On admet généralement que la capacité de cette mémoire est limitée et ne permet de traiter qu'un nombre limité d'items, variant selon les individus de 5 à 9, selon la formule "magique" de MILLER (1975) :  $7 \pm 2$ . *La mémoire à long terme* est constituée par une organisation complexe de connaissances, composée de schémas cognitifs. Sa capacité est pratiquement illimitée. Bien entendu, il peut survenir un phénomène d'oubli ou de déformation de certaines informations, celui-ci n'étant pas dû à la capacité de la mémoire humaine, mais à l'organisation et à l'encodage de ces informations dans la mémoire.

Exploitant les recherches menées en psychologie cognitive, plusieurs modèles IA, symboliques ou connexionnistes, de la mémoire humaine ont été réalisés. Le plus connu est sans doute celui de CARD et al. (1983), appelé "The Model Human Processor" (figure 1.10), qui s'intéresse surtout aux aspects temporels de



l'interaction homme-machine, dans le but de comprendre et de prévoir les performances du système homme-machine.

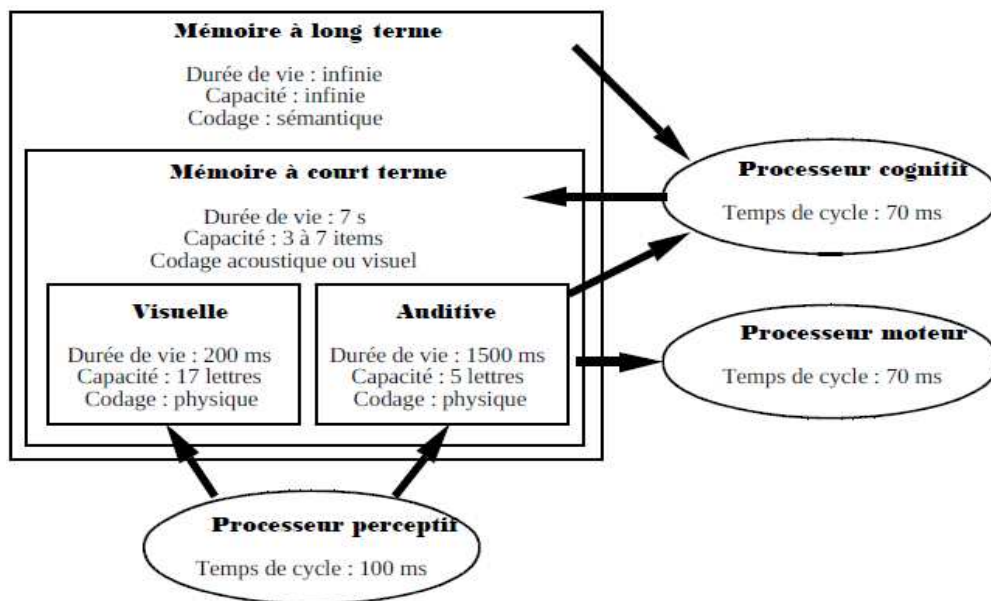


Figure 1.10 : Un modèle de la mémoire humaine (selon CARD et al., 1983)

Selon ces auteurs, l'utilisateur d'un logiciel peut être représenté par un système de traitement d'informations régi par des règles, et qui comprend un ensemble de mémoires et de processeurs interconnectés. Chaque mémoire est caractérisée particulièrement par trois paramètres : sa capacité, sa persistance (ou durée de vie) et le type d'information mémorisée. Les informations stockées sont exploitées et mises à jour par des processeurs extérieurs. Ensuite, chaque tâche peut être décomposée en opérations élémentaires. Ces opérations sont ensuite mises en relation avec les besoins potentiels humains en matière de mémorisation, ce qui conduit à identifier des besoins en assistance.

Dans les sciences cognitives, cette approche de modélisation est progressivement abandonnée au profit de la notion de mémoire de travail.

#### 1.2.2.1.2. Seconde approche de modélisation sous la forme d'une mémoire de travail

La notion de mémoire de travail a été précisée par les travaux de BADDELEY et HITCH (1974) : ceux-ci ont montré qu'un sujet peut maintenir un certain nombre d'items en mémoire à court terme (jusqu'à six éléments) tout en réalisant des opérations cognitives complexes telles que des opérations d'apprentissage, de compréhension ou encore la mise en oeuvre de raisonnements. Ces résultats ne sont donc pas compatibles avec le modèle précédent.

Cette constatation a conduit à la proposition d'un ensemble de modèles, symboliques ou connexionnistes. Par exemple, un des plus connus est le modèle symbolique ACT\* (Adaptive Control of Thought) proposé par ANDERSON (1983), et résumé en figure 1.11. La mémoire de travail, où sont représentées la situation courante et l'issue des traitements, est en relation avec une mémoire permanente déclarative et une mémoire permanente de nature procédurale :

- la mémoire permanente déclarative stocke des informations factuelles ou conceptuelles dans un réseau sémantique et sous un format variable, par exemple des images, des séquences ou des propositions ;
- la mémoire permanente de nature procédurale stocke des procédures dont un modèle peut être des règles de production. Ces procédures sont ensuite appliquées aux contenus de la mémoire de

travail. Ces applications peuvent engendrer de nouvelles connaissances déclaratives, créer d'autres règles de production ou modifier d'anciennes règles.

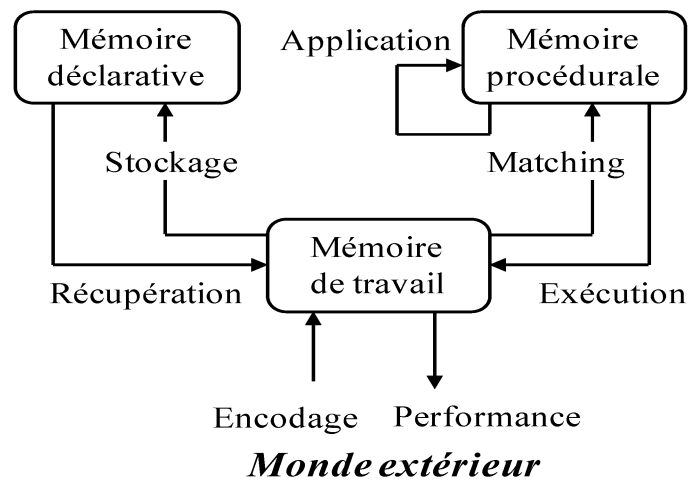


Figure 1.11. Le modèle ACT\* (selon ANDERSON, 1983)

On retrouve désormais ces principes de modélisation dans la plupart des modèles actuels. Des exemples seront donnés plus loin (cf. 1.2.2.3).

#### 1.2.2.1.3. Caractéristiques de la mémoire humaine et ergonomie des logiciels

Quelle que soit l'approche de modélisation utilisée, les limitations liées aux caractéristiques de la mémoire humaine ont des répercussions directes sur l'ergonomie des interfaces homme-machine et des outils d'aide mis à la disposition de leurs utilisateurs :

- D'une manière générale, une interface homme-machine est conçue comme un prolongement de la mémoire à court terme de l'utilisateur (BOY, 1988a).
- De plus, elle peut faire office de mémoire à long terme annexe de l'utilisateur, par exemple en centralisant et en rendant facilement disponibles les procédures de correction ou de reprise de défaut utilisables suite à l'apparition d'un dysfonctionnement dans le système.

A partir de ces constatations, de nombreuses recommandations ergonomiques peuvent être formulées, par exemple : "l'interface ne doit pas présenter des informations inutiles à la tâche en cours de réalisation, car elles limitent la capacité de travail et l'efficacité de l'utilisateur". De telles recommandations peuvent être trouvées parmi de centaines d'autres dans les guides d'ergonomie des logiciels mis de plus en plus à la disposition des concepteurs, cf. 1.4.2.

#### 1.2.2.2. Tentatives de modélisation des activités humaines

Les deux approches de modélisation des activités humaines les plus couramment considérées actuellement sont certainement celles proposées par RASMUSSEN (1980) et NORMAN (1986), représentatives de deux écoles de pensée, l'une européenne, et l'autre américaine. Malgré leurs limites, elles ont permis à de nombreux développeurs de systèmes interactifs de prendre conscience d'un ensemble de notions, issues de la psychologie, relatives aux interactions homme-machine.

### 1.2.2.2.1. Modélisation de la résolution humaine d'un problème

En 1980, RASMUSSEN a été le premier à proposer un cadre, appelé "échelle de décision" (decision ladder), illustrant la démarche générale de résolution de problème suivie par un utilisateur d'imagerie en salle de supervision. "L'échelle de décision" de RASMUSSEN comprend plusieurs étages séquentiels de traitement d'informations, figure 1.12 : la détection d'événement anormal met l'utilisateur en état d'alerte, suite à l'apparition d'une alarme ou par l'observation de l'évolution anormale d'une ou plusieurs variables ; l'évaluation de la situation consiste pour l'utilisateur à observer l'ensemble des informations utiles de façon à identifier l'état du système ; compte tenu de l'état précédemment identifié et des objectifs assignés à l'utilisateur, celui-ci définit une stratégie générale de correction, qu'il décompose ensuite en tâches, puis en procédures d'actions ; le dernier étage concerne l'exécution des actions.

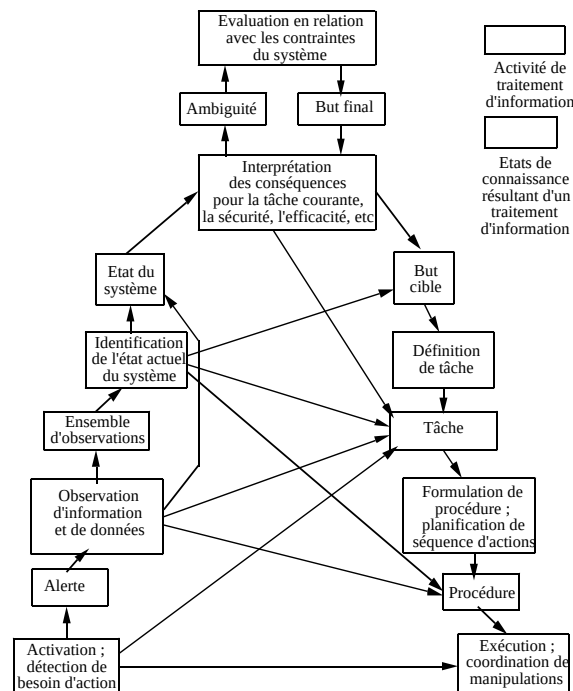


Figure 1.12 : "L'échelle de décision" (RASMUSSEN, 1980)

"L'échelle de décision" offre un bon cadre pour analyser le comportement cognitif de l'homme et proposer en conséquence une interface homme-machine adaptée à chaque de comportement suivi. A ce sujet, à partir de la modélisation proposée, RASMUSSEN (1983) distingue trois types de comportements, discutés ici par rapport à l'utilisateur d'un système interactif dans une salle de supervision de système complexe, figure 1.13. Le premier comportement correspond à celui basé sur des habilités, sur l'expérience, où l'utilisateur effectue de façon quasi-automatique des actions correctrices en réponse à des informations perçues sous forme de signaux, figure 1.14. Ceux-ci donnent des indications sur la situation courante de l'environnement, pour laquelle l'utilisateur expérimenté associe directement différentes actions. Il passe alors de l'étape d'activation/détection à l'étape d'exécution d'une procédure, figure 1.12.

Le second comportement est fondé sur des règles, où l'utilisateur, lorsqu'il est face à une situation familière ou préplanifiée, applique un ensemble de règles pour identifier l'état du système et choisir la procédure de reprise appropriée. D'après BOY (1986), les connaissances d'un expert extraites pour remplir la base de connaissance d'un système expert sont essentiellement de ce type. Le troisième comportement est basé sur la connaissance et est suivie lorsque l'utilisateur est face à des situations nouvelles ou imprévues. Il suit alors toutes les étapes de la démarche décisionnelle : il adapte alors des raisonnements réellement intelligents, émet des hypothèses, tente de les vérifier et de prédire l'évolution du système suite à d'éventuelles actions de correction. Les procédures appliquées seront mémorisées par l'utilisateur qui adoptera ensuite un comportement basé sur des règles s'il rencontre cette situation ultérieurement.

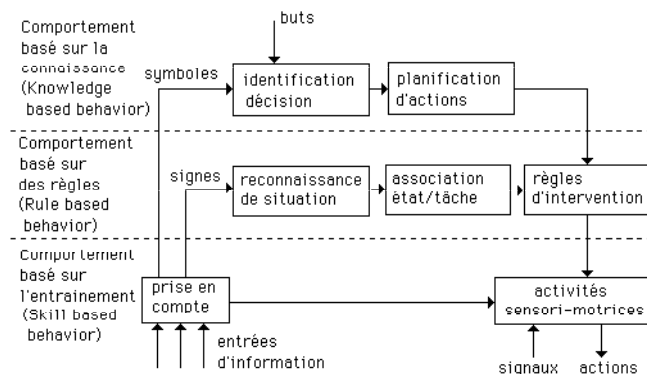


Figure 1.13 : Les trois types de comportement identifiés par RASMUSSEN (1983)

On retrouve implicitement, dans cette tentative de modélisation, *la notion d'économie cognitive*. En effet, elle laisse bien supposer que l'expérience acquise conduit à une organisation des connaissances permettant, dans les situations connues ou routinières, l'utilisation de connaissances bien adaptées, et donc une moindre charge cognitive (FALZON, 1989). Ainsi, pour chacune des situations rencontrées, et en fonction des outils d'assistance mis à sa disposition, l'utilisateur adoptera le comportement le moins coûteux possible, et donc utilisera les connaissances les plus adaptées possibles à la réalisation de la tâche en cours.

"L'échelle de décision" de RASMUSSEN a d'abord été considérée comme un modèle, puis de plus en plus comme une architecture cognitive générique, produisant une classe de modèles (HOC, 1993). Ce cadre a souvent été critiqué pour ses aspects réducteurs dans la mesure où il ne reflète pas la dynamique de la résolution de problème, en étant trop linéaire, trop figé. Notons à ce sujet que des recherches actives sont menées actuellement par HOC et AMALBERTI dans le but de proposer une modélisation (plus réaliste d'un point de vue temporel que celle de RASMUSSEN) du diagnostic et de la prise de décision dans les situations dynamiques (situations du type de celles qui nous intéressent dans ce mémoire). C'est ainsi qu'un modèle représenté sur une échelle de temps a été proposé récemment par ces auteurs (HOC, 93 ; HOC et AMALBERTI, sous presse). Dans ce modèle, les auteurs visent à faire apparaître trois dimensions : (i) la dimension "parallélisme/séquentialité" de l'activité humaine, (ii) la dimension "autonomie/dépendance" entre le diagnostic et ses conséquences, où par exemple le diagnostic pourra se traduire par une restriction des hypothèses à celles qui sont compatibles avec les actions disponibles, (iii) la dimension "diagnostic immédiat/différé", où le diagnostic n'est pas seulement une activité réactive, dans la mesure où un diagnostic peut aussi être anticipé. D'après HOC et AMALBERTI, ce modèle est susceptible de générer une classe plus large de modèles. Ce modèle nous paraît très prometteur ; c'est avec beaucoup d'intérêt que nous attendons sa validation sur le terrain et/ou en simulation.

#### 1.2.2.2.2. La théorie de l'action de NORMAN (1986)

La théorie avancée par NORMAN introduit la notion de modèle conceptuel et explique les différentes étapes cognitives nécessaires à la réalisation d'une tâche exécutée à l'aide d'un système informatique. Le modèle conceptuel correspond à une représentation mentale en terme de variables dites psychologiques : à chaque concept, unité de connaissance ou sujet d'intérêt correspondra une variable psychologique. La théorie de l'action comporte sept étapes :

- *L'établissement d'un but* : un but est une représentation mentale de l'état du système que l'utilisateur souhaite atteindre, et ceci en agissant sur des dispositifs de commande.
- *La formulation d'une intention* : celle-ci correspond à la décision d'agir afin d'atteindre un but pré-établi.

- *L'adoption par l'utilisateur d'un plan d'action* : elle correspond à la représentation psychologique de l'ensemble des actions et de leur ordonnancement que l'utilisateur doit exécuter au moyen des dispositifs physiques, et ceci dans le but d'atteindre son objectif.
- *L'exécution du plan d'action* : le plan est mis en application en agissant sur le système. Celui-ci est donc modifié.
- *La perception du nouvel état du système* : l'utilisateur constate un ensemble de changements survenus sur le système.
- *L'interprétation de la modification des variables physiques en terme psychologique* : elle aboutit à une représentation mentale du nouvel état du système.
- *La comparaison entre l'état du système avec le but pré-établi et les intentions formulées* : elle peut conduire à la poursuite du plan ou à sa modification.

En fait, l'activité de l'utilisateur ne se ramène pas à une simple exécution séquentielle de ces étapes : ces dernières peuvent apparaître dans un ordre différent, certaines peuvent ne pas être exécutées, et d'autres peuvent être répétées. Ainsi, il est maintenant instructif de discuter de cette théorie dans le contexte qui nous préoccupe, en l'occurrence la réalisation de tâches humaines dans une salle de supervision de système complexe, figure 1.14. En cas d'apparition d'un défaut, l'utilisateur interprète et évalue la situation de fonctionnement du système. Puis, en fonction des buts opératoires, il traduit ses intentions en une suite d'actions qu'il exécute. On distingue donc bien dans cette interaction les variables physiques, correspondant à des variables du système (température, pression...) et les variables psychologiques à partir desquelles l'utilisateur évalue la situation de fonctionnement. L'utilisateur est par conséquent amené à traduire dans un premier temps les variables physiques en variables psychologiques et cela dans un but d'évaluation de la situation. Dans un second temps, la traduction des variables psychologiques en variables physiques lui permet de spécifier les actions à mettre en œuvre. La phase d'évaluation consiste pour l'utilisateur à comparer l'état du système avec les buts et les intentions d'origine. Le problème de l'interaction est donc bien posé en termes de compatibilité entre le modèle de l'utilisateur et l'image du système.

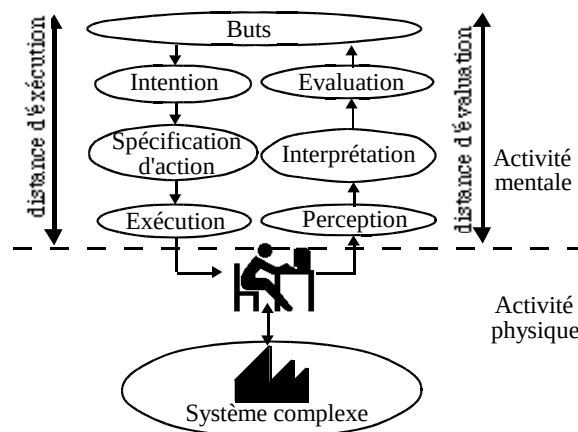


Figure 1.14 : Les étapes de la théorie de l'action dans un contexte de supervision de système complexe (adapté de NORMAN, 1986)

Cette approche de modèle met en évidence le rôle primordial de l'interface homme-machine. En effet, tout au long de son activité, l'utilisateur se construit un monde virtuel et ceci, par l'intermédiaire de l'interface doit donc être une image du monde réel. De plus, l'interface doit faciliter la mise en correspondance entre les variables psychologiques du monde virtuel et les variables physiques du monde

réel ainsi que l'établissement d'associations entre les variables physiques et les dispositifs de supervision. Cette mission est confiée au concepteur qui doit en conséquence établir explicitement ces relations, gages de qualité en définitive pour l'interface homme-machine.

#### 1.2.2.2.3. Modélisation des activités humaines et spécification des interfaces et des modules d'aide

Ces approches de modélisation constituent une source précieuse d'enseignement et de réflexion pour les concepteurs des interfaces homme-machine et des outils d'assistance. En effet, elles permettent de mieux positionner et comprendre les besoins de l'utilisateur pour les différentes tâches qu'il a à accomplir, et ceci dans les différentes situations, normales et anormales, du système. Par exemple, en s'inspirant de la figure 1.12, il est possible de mettre en évidence sur le modèle de résolution de problème proposé par RASMUSSEN les différents types de modules d'aide qui pourront éventuellement être utiles (sinon nécessaires) à l'utilisateur pour réaliser ses tâches. Inspirée de MILLOT (1988), la figure 1.15 donne un aperçu *non exhaustif* des types de modules possibles en fonction de l'étape dans laquelle l'utilisateur se trouve. Outre un module d'acquisition de données, différents modules peuvent être envisagés : (i) un module de gestion d'alarme, (ii) un module permettant la visualisation de données issues du système, (iii) un module de diagnostic activable en cas de défaillance, (iv) un module capable de prédire l'état futur probable du système, (v) un module permettant de simuler l'état du système suite à une action sur une des variables, (vi) un module capable de donner des conseils d'action à l'utilisateur, enfin, (vii) un module permettant d'effectuer des commandes sur le système.

D'autres précieux enseignements pourront être tirés des tentatives d'informatisation de modélisations cognitives.

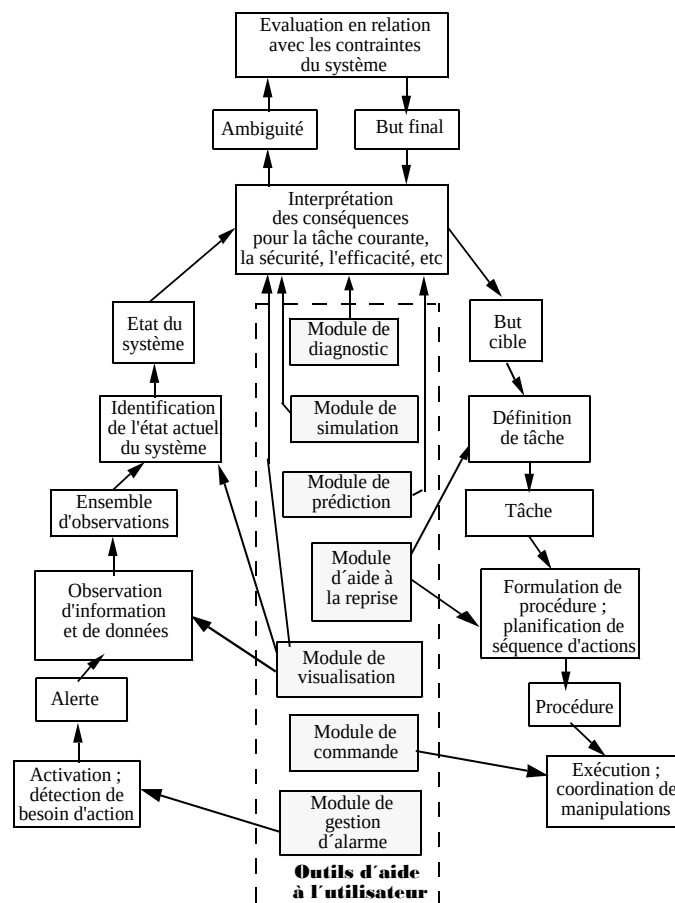


Figure 1.15 : Positionnement de modules d'aide à l'utilisateur en fonction de la tâche

### **1.2.2.3. Tentatives d'informatisation de modélisations cognitives**

Les premiers modèles informatisés simulaient le comportement humain pour un type de tâches données. Ils étaient qualifiés de modèles normatifs par BOY (1983), car la performance de l'être humain simulé était définie par une norme spécifique d'un mode de fonctionnement donné du système homme-machine. Mais, comme l'explique MILLOT (1988) dans une synthèse sur les différents types de modélisation, ces modèles se basaient le plus souvent sur des hypothèses simplificatrices concernant le comportement décisionnel global et les stratégies opératoires. Malgré leurs aspects simplificateurs impliquant un faible niveau cognitif, ces premiers modèles étaient un premier pas pour étudier et améliorer l'ergonomie des systèmes homme-machine en prédisant leurs points sensibles.

Une seconde classe de modélisation, plus récente, regroupe les modèles dits psycho-explicatifs (BOY, 1983). En partant d'une analyse de tâche, elle vise à simuler les processeurs humains d'acquisition et de traitement d'information, de décision et d'action lors de l'exécution de la tâche, en bref à simuler le comportement humain dans des tâches de résolution de problème. Les stratégies opératoires sont le plus souvent stockées dans une base de connaissance. Le lecteur trouvera dans BERSINI (1989) un ensemble de techniques issues principalement de l'intelligence artificielle, contribuant à la modélisation cognitive dans l'étude des systèmes homme-machine. On retrouve dans la littérature plusieurs applications impliquant de tels modèles. Par exemple, MESSAGE (BOY et TESSIER, 1985) modélise le comportement du pilote et du copilote d'un Airbus A310. KARL (KNAEUPER et ROUSE, 1985) et CES (WOODS et ROTH, 1987) simulent des prises de décision humaines dans des tâches de résolution de problème en salle de supervision.

Mais la tentative de simulation d'un modèle cognitif la plus récente est sans doute celle de COSIMO. Celui-ci est issu de concepts cognitifs recensés par REASON (1987), et développé au centre de recherche d'Ispira en Italie (CACCIABUE et al., 1990b). Ce projet a pour objet de modéliser et de simuler, par un programme informatique, les caractéristiques cognitives d'un opérateur humain jugées importantes dans la genèse de certaines erreurs humaines, ainsi que des incidents et des accidents (MASSON, 1994). L'architecture comprend deux niveaux cognitifs de raisonnement et de prise de décision : (i) le niveau "haut" doit permettre d'exploiter les connaissances humaines en reconnaissant, en continu, les différentes situations (diagnostic) et en construisant des stratégies de supervision (planification), (ii) le niveau "bas" vise l'accomplissement d'une réponse préprogrammée ou d'une stratégie planifiée dans le but de satisfaire une intention clairement définie. Il s'agit ensuite pour le modèle d'exécuter et d'optimiser la stratégie sélectionnée. Les concepts sur lesquels repose le modèle sont au niveau perceptif la "salience physique" et "la salienc cognitive", et au niveau de la résolution de problème les concepts de "sélection par la fréquence" et "d'appariement par similarité" (REASON, 1987). Le modèle COSIMO est implanté sur une machine LISP. Une architecture "Blackboard" (NII, 1986a ; 1986b) permet de simuler l'ensemble des activités cognitives dans une structure informatique intégrée. Cette structure permet de passer d'une activité cognitive à l'autre de façon opportuniste lors de la résolution d'un problème. Bien entendu, le modèle n'est pas terminé mais il constitue une avancée importante dans ce domaine.

En conclusion, les travaux menés sur les tentatives de modélisation cognitive doivent être suivis avec beaucoup d'attention par les concepteurs d'interfaces homme-machine. En effet, d'après CACCIABUE et al. (1990a), ces travaux peuvent infléchir la conception des systèmes et des interfaces, particulièrement pour des systèmes qui n'existent pas encore et qui ne peuvent pas être observés, et ainsi contribuer à la conception d'aides intelligentes à la décision. Ils permettent également de progresser dans la compréhension des mécanismes et des stratégies cognitives dans des environnements complexes.

### **1.2.3. Conclusion sur l'analyse et la modélisation des tâches et de l'utilisateur**

Le génie logiciel est particulièrement riche en méthodes et en modèles. Cependant, des lacunes apparaissent clairement dès lors que le développement concerne un système interactif. En effet, mises à part

quelques exceptions, l'analyse et la modélisation des tâches humaines et de l'utilisateur sont très souvent laissées à l'appréciation et à l'expérience des concepteurs. Comment s'étonner alors que tant de projets de développement d'applications interactives aient conduit à des échecs partiels ou totaux depuis une vingtaine d'années ?

L'apparition de méthodes de spécification de plus en plus variées et d'outils de réalisation de plus en plus puissants et simples d'utilisation n'est pas suffisante. Il s'agit encore d'intégrer systématiquement et explicitement la prise en compte des facteurs humains dans les méthodes de développement, et d'inculquer aux analystes et concepteurs de nouvelles méthodes de travail. Avant de recenser les outils de spécification et de réalisation d'interface homme-machine, il est utile de s'intéresser de plus près aux différents modèles d'architectures possibles pour celle-ci.

### 1.3. MODELES D'ARCHITECTURE DE L'INTERFACE HOMME-MACHINE

Au début des années 80, suite à de nombreux problèmes relatifs au manque de productivité lié au développement des premiers systèmes interactifs, et en essayant de pallier certaines faiblesses en terme d'ergonomie d'utilisation, des recherches ont conduit à définir un système interactif comme l'assemblage de deux composants logiciels indépendants, prenant en charge séparément les aspects liés à l'application et à l'interface homme-machine (figure 1.16).

D'après JURAIN (1991), les grosses applications posant des problèmes de flexibilité, de maintenance et d'adaptabilité, il s'agit de séparer le traitement des données de la gestion des informations échangées avec l'utilisateur, et ceci afin de répartir les tâches de programmation entre les concepteurs d'application et les concepteurs des interfaces. Le développement d'une part de l'application, et d'autre part des interfaces, nécessite des développeurs aux compétences complémentaires. L'importance accordée à l'interface homme-machine suite à cette séparation nette a contribué à faciliter un premier rapprochement entre le développeur et l'utilisateur. En effet, cette séparation permet particulièrement de tester les interfaces d'une maquette d'application sans avoir à développer complètement les fonctionnalités de celle-ci, de prendre en compte les remarques des utilisateurs, puis de tester de la même manière les interfaces lors du passage au prototype.

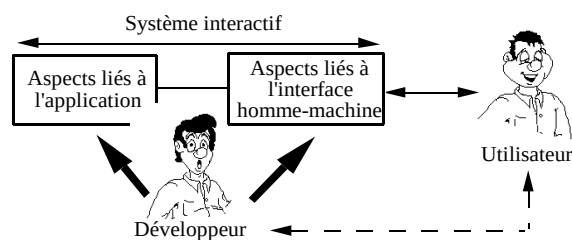


Figure 1.16. Séparation de l'interface homme-machine en deux composants logiciels distincts

On assiste donc à l'avènement d'un ensemble d'architectures proposant une séparation entre application et interface homme-machine. Plusieurs modèles représentatifs seront d'abord décrits. Ensuite, il sera plus facile de positionner des concepts d'interfaces dites "intelligentes" ou évoluées.

#### 1.3.1. Présentation de quatre modèles représentatifs

On distingue dans la littérature de nombreux modèles d'architecture. Sans souci d'exhaustivité, mais plutôt de représentativité, et en s'inspirant des études à ce sujet de COUTAZ (1990) et de DUVAL (1993), seront présentés successivement les principes de quatre modèles parmi les plus cités et étudiés actuellement. Les trois premiers appartiennent à "la famille SEEHEIM". Le quatrième modèle est MVC, lié à Smalltalk. Ils constituent une source considérable d'idées.



### 1.3.1.1. Le modèle *LANGAGE*

Le modèle *LANGAGE* a été proposé par FOLEY et VAN DAM en 1982. Ce modèle part sur l'hypothèse que l'application et l'interface utilisateur peuvent se décomposer de la même façon qu'un langage. Cette décomposition s'effectue selon quatre niveaux (figure 1.17) :

- *Le niveau conceptuel* : à ce niveau, les objets de l'application sont recensés, ainsi que leurs attributs, leurs relations et les actions qu'il est possible d'effectuer sur ces objets. Ce niveau correspond à l'application.
- *Le niveau sémantique* : on y retrouve la description des fonctionnalités du système, en précisant l'ensemble des actions possibles sur les différents objets de l'application, sans se préoccuper des liens séquentiels entre actions.
- *Le niveau syntaxique* : on y définit les séquencements des actions de l'utilisateur en termes d'entrées/sorties, et leur format afin de réaliser les fonctionnalités spécifiées au niveau supérieur. Ce niveau doit donc se baser sur un vocabulaire de description et une grammaire.
- *Le niveau lexical* : on y retrouve des primitives permettant de gérer les événements d'entrée et d'assurer l'affichage des sorties.

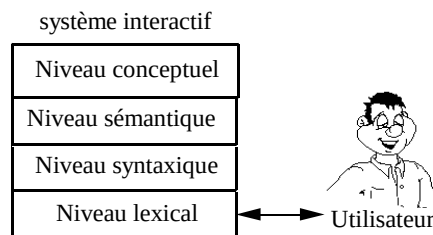


Figure 1.17 : Modèle *LANGAGE*

Les principes de base du modèle ont été exploités par de nombreux chercheurs, et ont influencé la définition de plusieurs modèles génériques. Le plus connu est sans doute le modèle de *SEEHEIM*.

### 1.3.1.2. Le modèle de *SEEHEIM*

Au milieu des années 80, un groupe de travail *SIGGRAPH* définit un modèle d'interaction homme-ordinateur appelé modèle de *SEEHEIM* (PFAFF, 1985), prenant son nom du lieu où il a été défini, à *SEEHEIM* en RFA à l'occasion d'un séminaire sur les systèmes de gestion des interfaces utilisateur. Ce modèle comprend trois composants logiques (figure 1.18) :

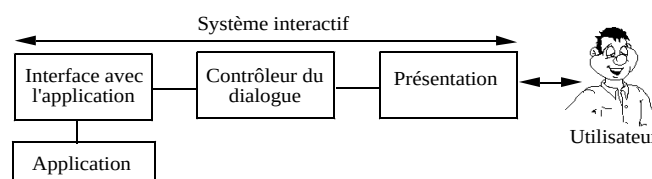


Figure 1.18 : Modèle de *SEEHEIM*

- *La présentation* est la partie visible par l'utilisateur, et gère les entrées/sorties. Par rapport au modèle *Langage*, elle correspond globalement au niveau *lexical*.

- *Le contrôleur du dialogue* est responsable de la structure du dialogue entre l'utilisateur et le système interactif. Il peut être vu comme le niveau *syntactique* du modèle LANGAGE. Ainsi, comme l'explique COUTAZ (1990), les phrases syntaxiquement correctes correspondent à des requêtes et des données que l'utilisateur souhaite transmettre à l'application, et dans le sens inverse, le contrôleur reçoit des phrases de sortie abstraites qui sont ventilées vers les éléments spécialisés de la présentation.
- *L'interface avec l'application* est chargée de la communication des données entre l'interface et l'application, et ceci tout en respectant la sémantique des données et du dialogue. Elle correspond au niveau *sémantique* du modèle Langage.

Remarquons que l'application correspond au niveau *conceptuel* du modèle Langage.

D'après COUTAZ (1990), ce modèle, bien que possédant des limites liées à sa généralité, offre plusieurs avantages, tels que (i) proposer un cadre de pensée pour jeter les bases structurelles d'un système à construire, (ii) faciliter la conception itérative des interfaces dans la mesure où il est possible de modifier la partie présentation indépendamment des autres, (iii) ne pas être lié à une technique particulière de réalisation, (iv) servir de point de départ à la proposition d'outils de génération automatique d'interface spécialisés sur chacun des trois niveaux de description (lexical, syntaxique et sémantique). D'après BEAUDOUIN-LAFON (1991), l'interprétation linguistique du modèle de SEEHEIM est correcte pour décrire un dialogue de haut niveau, mais n'est pas appropriée si l'on veut concevoir des applications à manipulation directe.

Ce modèle d'architecture est la référence pour les applications interactives. Il a fait l'objet de dérivés, tels le modèle de SEEHEIM modifié (DANCE et al., 1987) visant à étendre le niveau sémantique de l'architecture, le modèle de SEEHEIM étendu (KARSENTY et WEIKART, 1991) qui vise à inclure complètement l'application dans le modèle par des composants supplémentaires, ou encore le modèle ARCH (BASS et al., 1991) qui divise une application interactive en cinq composants : boîte à outils et présentation (correspondant à une décomposition physique et logique de la présentation), dialogue, adaptateur de domaine et domaine. Le modèle PAC présenté ci-dessous est également inspiré du modèle de SEEHEIM.

### 1.3.1.3. Le modèle PAC

Le modèle PAC (Présentation Abstraction Contrôle), proposé par COUTAZ (1987 ; 1990), part sur l'hypothèse qu'une application et son interface homme-machine peuvent être décomposées en une hiérarchie d'agents interactifs. Les niveaux sémantiques, syntaxiques et lexicaux se retrouvent donc distribués dans des entités de base appelées objets interactifs. Chaque entité se compose de trois composants (figure 1.19) :

- *Le composant Présentation* : celui-ci définit la syntaxe concrète de l'application, c'est-à-dire l'aspect visuel des objets, de même que les comportements d'entrée/sortie sur les objets, et ceci vis-à-vis de l'utilisateur.
- *Le composant Abstraction* : il désigne la sémantique, c'est-à-dire les fonctionnalités de l'application.
- *Le composant Contrôle* : celui-ci maintient la cohérence entre les aspects de présentation et d'abstraction. Il fait office d'arbitre et de centre de décisions, et décide des techniques de traduction entre les formes abstraites et concrètes d'un objet PAC.

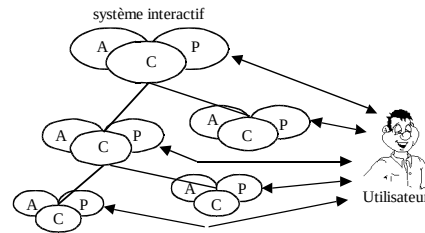


Figure 1.19 : Structuration d'un système interactif en objets PAC

Le modèle PAC correspond en fait à une répartition distribuée des composants du modèle SEEHEIM, en s'appuyant sur la notion d'agent. En ce sens, il peut bénéficier des avantages de cette notion : grande modularité, perception de l'application à différents niveaux d'abstraction, et parallélisation éventuelle des entrées/sorties. Plusieurs recherches visent actuellement à exploiter au maximum les intérêts d'une telle approche d'architecture répartie, par exemple :

- COUTAZ et NIGAY (1991) proposent un modèle hybride appelé PAC-SEEHEIM, combinant le modèle de SEEHEIM et les principes multi-agent de PAC. En particulier, le contrôleur de dialogue du modèle de SEEHEIM peut être composé d'agents PAC.
- DUVAL (1993) combine les principes des modèles de SEEHEIM, PAC et ARCH pour proposer un modèle unifié appelé SPA. Un système interactif se compose alors d'une hiérarchie d'objets SPA, composés chacun de six composants : présentation, interface présentation, contrôle, interface contrôle, application et interface application.

#### 1.3.1.4. Le modèle MVC

Contrairement aux modèles précédents qui restent pour l'instant conceptuels, le modèle MVC (Modèle, Vue, Contrôleur) est le modèle d'architecture logicielle du langage orienté objet Smalltalk, où toute entité est un objet (GOLDBERG, 1984). Ce modèle permet de gérer la cohérence globale d'une application interactive en assurant la cohérence entre les différentes vues d'un même objet.

Cette responsabilité est partagée entre trois objets (figure 1.20) : (i) *le modèle* réunit l'ensemble des fonctions abstraites, et représente les structures de données de l'application, (ii) *la vue* définit la présentation en sortie, en gérant les tâches graphiques, (iii) *le contrôleur* est chargé de la gestion des interactions avec l'utilisateur, en étant capable d'interpréter les entrées en provenance de l'utilisateur.

En comparant le modèle MVC à PAC, COUTAZ (1990) met en évidence trois différences fondamentales : (i) MVC au contraire de PAC est très lié aux techniques de programmation par objets, (ii) MVC utilise les objets Vue et Contrôleur pour réaliser la notion équivalente de Présentation dans PAC, et enfin (iii) dans MVC le contrôle au sens PAC est inexistant et ses fonctions sont diluées entre le modèle, la vue et le contrôleur. Il est important de souligner qu'une mise en oeuvre rigoureuse du modèle MVC exige un long apprentissage de la programmation objet en Smalltalk.

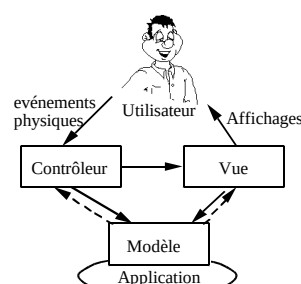


Figure 1.20 : Modèle MVC

### 1.3.1.5. Conclusion sur les modèles décrits

Les principes de base de quatre modèles d'architectures d'interfaces, LANGAGE, SEEHEIM, PAC et MVC, ont été décrits. Ils constituent une avancée très importante dans l'étude et la mise en oeuvre des systèmes interactifs. Cependant, ils ne s'intéressent qu'aux aspects purement logiciels de l'interface. Ainsi, ils répondent à un ensemble de critères du génie logiciel, par des aspects de modularité ou encore de réutilisabilité, mais ne prennent pas en compte explicitement certains aspects pourtant fondamentaux des interactions homme-machine, liés aux facteurs humains.

Au vu des progrès technologiques réalisés au niveau des aspects graphiques interactifs, l'architecture logicielle des systèmes interactifs peut progressivement s'organiser autour de *l'intégration de l'utilisateur*. Nous avons vu précédemment que celui-ci doit être beaucoup plus considéré comme un composant à part entière du système homme-machine et, que dans le cycle de développement du système interactif, de nouvelles notions doivent apparaître : les tâches à réaliser par l'utilisateur, les objectifs à atteindre, la prise en compte de l'environnement de travail, les connaissances nécessaires pour utiliser le système, l'ergonomie de présentation d'information, etc, conduisant progressivement à des aspects de *modélisation de l'utilisateur et des tâches*.

C'est pourquoi des recherches visent actuellement à proposer des architectures d'interface et les tâches intégrant explicitement des connaissances sur l'utilisateur. Même si ces recherches n'en sont qu'à leurs premiers balbutiements en raison de leurs objectifs ambitieux, des concepts et des modèles particulièrement prometteurs commencent à voir le jour. Ils font l'objet de la partie suivante.

### 1.3.2. Positionnement de concepts d'interfaces "intelligentes" dans le modèle de SEEHEIM

Actuellement, certaines recherches visent l'intégration de facteurs humains directement dans l'interface en s'orientant progressivement vers des structures d'interfaces dites "intelligentes" ou évoluées, se chargeant exclusivement du problème de dialogue entre l'utilisateur et l'ensemble des partenaires du système homme-machine. Plusieurs d'entre elles sont successivement présentées et précisées par rapport au modèle de SEEHEIM.<sup>7</sup>

#### 1.3.2.1. Interface flexible

Le concept d'interface flexible, qualifiée aussi de "configurable", de "personnalisable" ou "d'adaptable", vise à permettre l'utilisation d'un même support de dialogue pour communiquer avec plusieurs systèmes différents. Ainsi, l'utilisateur peut configurer l'interface et ses supports selon des critères personnels subjectifs : préférences dans les modes de présentation et/ou les styles d'interaction. Une interface flexible se compose généralement d'un ensemble de bases de données exploitées par un gestionnaire de présentation (WILLIGES et al., 1987). Ce dernier est piloté par l'utilisateur selon l'application visée.<sup>8</sup>

Les connaissances requises pour une interface flexible sont regroupées généralement dans trois

<sup>7</sup> Notons que l'appellation "intelligente" paraît à certains auteurs, tels COUTAZ et al. (1992), surfaite voire dangereuse dans la mesure où l'objectif n'est pas de demander au système l'apparence qui satisfait le test de Turing. Toutefois nous n'hésiterons pas à l'utiliser tout en ayant parfaitement conscience de cet état de fait, puisque les interfaces "intelligentes" constituent maintenant un domaine de recherche à part entière, des congrès et des ouvrages lui étant désormais consacrés.

<sup>8</sup> Les concepts d'interface flexible évoluent actuellement en prenant en compte progressivement des notions liées aux tâches humaines. C'est le cas par exemple du système MacWeb visant la génération d'interfaces adaptables (de type hypertexte) pour l'accès à une base de documents techniques. Dans ce système, l'accès aux informations contenues dans les documents est possible en fonction de leur utilisation par rapport à la tâche en cours, et non selon leur appartenance à un document particulier, par l'intermédiaire de liens décrits dans un modèle d'hypertexte à base de connaissance (NANARD et NANARD, 1993).

bases, figure 1.21 : (i) la première base contient l'ensemble des protocoles de communication avec les différentes applications constituant l'environnement de travail de l'utilisateur, (ii) la deuxième base contient les modes de présentation (couleurs, tailles de fenêtres, etc) et les différents styles d'interaction accessibles aux utilisateurs de l'interface, (iii) la troisième base, spécifique à chaque utilisateur du système global, contient les préférences de chacun d'entre eux dans les modes de présentation et/ou dans les styles d'interaction.

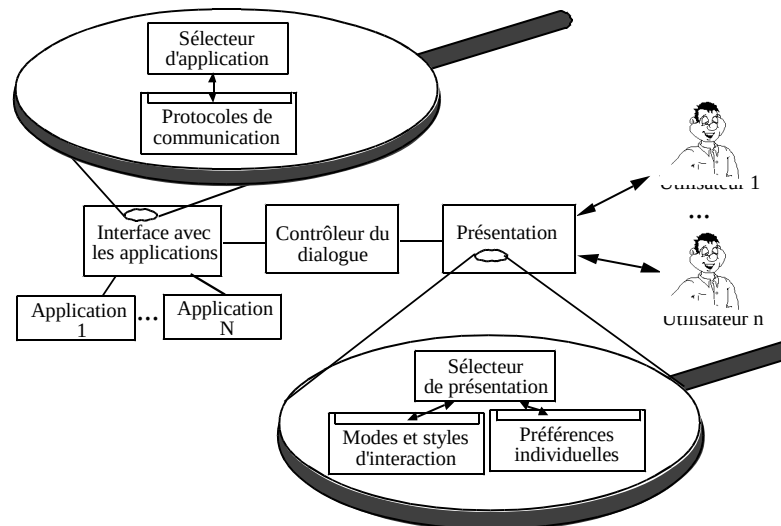


Figure 1.21 : Concepts d'interface flexible dans le modèle de SEEHEIM

Ainsi, en sélectionnant l'application, l'utilisateur sélectionne implicitement, grâce au sélecteur d'application, le protocole de communication adéquat. Il peut alors dialoguer avec l'application à travers une interface déjà configurée mais assez flexible pour qu'il puisse la reconfigurer de façon simple, et à tout moment. Ce type de système de dialogue offre plusieurs avantages : (i) les actions de l'utilisateur sont filtrées et traduites dans le langage de l'application sélectionnée, (ii) les informations provenant de l'application sont, de la même façon que les actions de l'utilisateur, filtrées et traduites selon un style prédéfini, (iii) l'utilisation d'un ensemble d'applications hétérogènes devient plus conviviale. Cependant, on peut mettre en évidence le coût de développement d'une telle interface, qui devient proportionnel au nombre d'applications et surtout à leur hétérogénéité.

### 1.3.2.2. Interface tolérante aux erreurs humaines

La mise en œuvre d'interface tolérantes aux erreurs humaines se base sur les résultats de travaux préalables consistant en l'observation d'utilisateurs dans des situations simulées. Ces observations visent à identifier leurs différentes erreurs se traduisant par des actions ne satisfaisant pas un ensemble pré-défini de critères. En situation réelle, le principe de ces interface consiste alors à remplacer, améliorer ou annuler les actions humaines non appropriées (ROUSE et MORRIS, 1985 ; HOLLNAGEL, 1989 ; MASSON et DE KEYSER, 1992 ; BEKA BE N'GUEMA et al., 1992). La structure d'une telle interface est généralement composée de trois modules principaux (figure 1.22).

Un module de décodage traduit les intentions de l'utilisateur (prenant la forme d'actions) en données exploitables par le système. Un second module classe d'abord l'action présumée de l'utilisateur, en se basant sur un modèle des tâches humaines prescrites et sur un modèle de l'application ; puis, il la corrige, dans le cas où elle n'est pas conforme aux consignes de production et/ou de sécurité. Un troisième module est chargé d'expliquer à l'utilisateur les problèmes liés à l'action d'origine et les avantages apportés par l'action corrigée, vis-à-vis des consignes de départ.

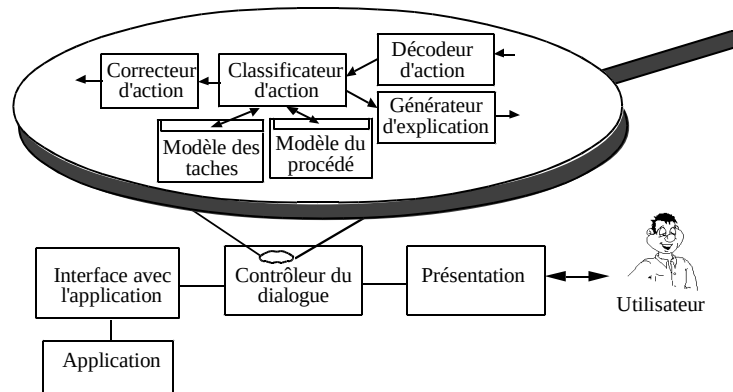


Figure 1.22 : Concepts d'interface tolérante aux erreurs humaines dans le modèle de SEEHEIM

L'utilité d'un tel principe est indéniable. Cependant, il n'est pas toujours applicable. En effet, un tel système doit se baser soit sur un modèle le plus fiable possible de l'application, soit sur des procédures de correction pré-établies à l'aide d'expériences préalables. Dans ce dernier cas, la liste des actions correctrices ne peut être exhaustive, car elle est généralement établie grâce à des scénarios de pannes bien définis. Mais, dans la réalité, il arrive que de nouvelles situations de pannes surviennent. Celles-ci sont donc méconnues de l'utilisateur et de l'interface tolérante aux erreurs humaines. Les risques d'erreurs deviennent alors importants, et une interface de ce type peut bien entendu s'avérer inefficace. Toutefois, d'après TENDJAOUI (1992), ce type de système de dialogue est particulièrement prometteur par sa capacité à s'adapter aux commandes de l'utilisateur.

### 1.3.2.3. Interface adaptative

Le concept d'interface adaptative est en général basé sur le fait que les modes de dialogue avec l'utilisateur doivent s'adapter aux tâches que celui-ci a à accomplir. De nombreuses architectures et méthodologies pour la conception d'interfaces ou d'outils adaptatifs existent dans la littérature (EDMONDS, 1981 ; ROUSE, 1988). Par exemple, synthétisant plusieurs approches, HEFLEY (1990) présente une architecture théorique fonctionnelle d'un système adaptatif "intelligent" reposant sur une modélisation de l'utilisateur. Elle est orientée vers un groupement d'agents dont le but est d'assurer les interactions entre une application donnée et son utilisateur (figure 1.23), et trace une séparation nette entre l'interface utilisateur et l'application. Elle intègre un modèle des tâches évolué sous forme d'un système "intelligent" d'aide à la décision qui correspond au modèle de l'application vue par l'utilisateur. Elle est composée des agents suivants :

- *L'identificateur des plans.* En liaison avec une base de connaissance, il infère des plans représentant des actions observées de l'utilisateur afin de déterminer leur but.
- *Le critique.* Cet agent est chargé d'identifier les actions des utilisateurs considérées comme critiques vis-à-vis des contraintes à respecter.
- *Le coordinateur* des connaissances sur le domaine est capable d'allouer certaines tâches de l'utilisateur au programme de l'application qui utilise des réponses planifiées. Pour cela, le coordinateur se base sur des heuristiques d'allocation de fonctions et sur des connaissances spécifiques à l'application.
- *Le générateur de réponses.* Cet agent se charge de répondre aux interrogations et aux questions de l'utilisateur, suite à des événements inconnus.

- *Le générateur d'explications.* Cet agent se charge de donner des explications sur les différentes conclusions du système d'aide. Ainsi, sur demande de l'utilisateur, cet agent justifie les raisons qui l'ont poussé par exemple à décharger l'utilisateur de telle tâche.
- *Le générateur de présentation.* Il doit assurer des modes de présentation de l'information sous une forme compatible avec l'état courant des connaissances et des besoins humains.

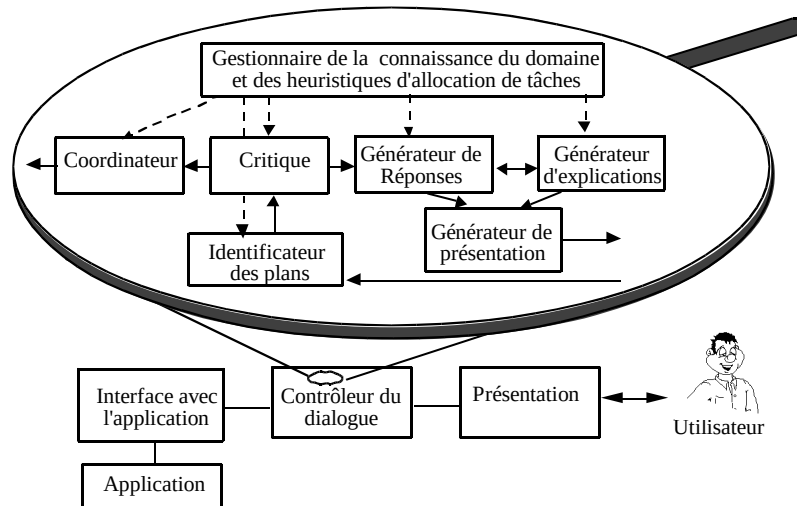


Figure 1.23 : Concepts d'interface adaptative dans le modèle de SEEHEIM

Ce type de système de dialogue peut être considéré comme une généralisation de l'interface tolérante aux erreurs humaines. Il fait l'objet de nombreuses recherches actuellement, mais reste très difficile à mettre en place surtout lorsque l'application est complexe.

#### 1.3.2.4. *Opérateur assistant*

Dans l'approche "d'opérateur assistant", l'interface se comporte comme un assistant pour l'utilisateur, et l'aide dans la réalisation de ses tâches. L'interface doit être capable d'initiatives personnelles, par exemple en testant des paramètres ou en prédisant des pannes déductibles, mais l'opérateur humain reste le décideur final. La structure de l'opérateur assistant est constituée de plusieurs modules, figure 1.24.

Les modules les plus importants sont : (i) le module de reconnaissance des situations qui permet, en fonction de la situation désirée, de la situation perçue et de connaissances situationnelles et analytiques placées dans une "mémoire à long terme", de sélectionner un sous-ensemble de règles à inférer ; (ii) le module de résolution de problème qui utilise le sous-ensemble de règles précédent, dans le but d'assister l'utilisateur, en répondant par exemple, à une requête particulière de celui-ci. Grâce à la structure de ses connaissances, l'opérateur assistant doit pouvoir disposer de plusieurs niveaux d'autonomie. Il peut être potentiellement considéré comme une généralisation de l'interface adaptative. Il fait l'objet de recherches actives visant une assistance intelligente en aéronautique et dans l'espace (BOY, 1988a, 1988b ; AMALBERTI et DEBLON, 1992).

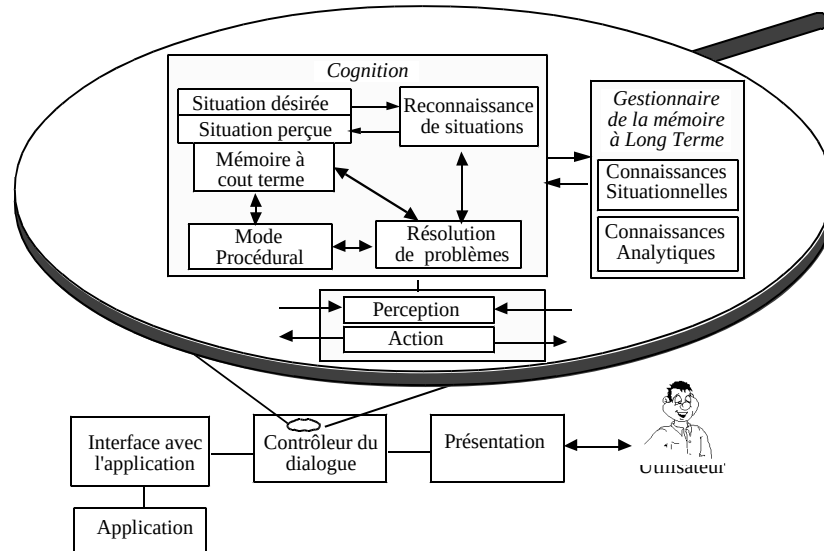


Figure 1.24 : Concepts d'opérateur assistant dans le modèle de SEEHEIM (inspiré de BOY, 1988b)

### 1.3.2.5. L'interface homme-machine vue comme un agent intelligent

Le concept d'agent intelligent est inspiré de travaux menés en Intelligence Artificielle Distribuée, tentant de modéliser par un formalisme logique le comportement d'un agent rationnel. Ce comportement est basé sur des états intentionnels, tels que les croyances, les désirs, les intentions et les engagements (CHAIB-DRAA, 1990 ; MANDIAU, 1993).

Le principe d'une interface, utilisant des concepts d'agent intelligent, repose sur la possibilité de décomposer le système homme-machine en un ensemble d'agents, figure 1.25. Ces agents travaillent en parallèle et/ou coopèrent, dans le but de résoudre différents problèmes relatifs aux tâches à effectuer. Le résultat de leurs traitements est transmis aux utilisateurs, grâce à des actes de communications, mais on peut envisager en fait tout un ensemble d'autres actions, sur le système par exemple. MANDIAU et al. (1991a ; 1991b) envisagent : (i) le système complexe sous forme d'agents réactifs, (ii) des systèmes d'assistance sous forme d'agents cognitifs, (iii) le groupe d'utilisateurs sous forme d'agents dits externes et (iv) d'autres interfaces vues comme des agents intelligents. L'interface homme-machine est donc intégrée dans un univers multi-agents.

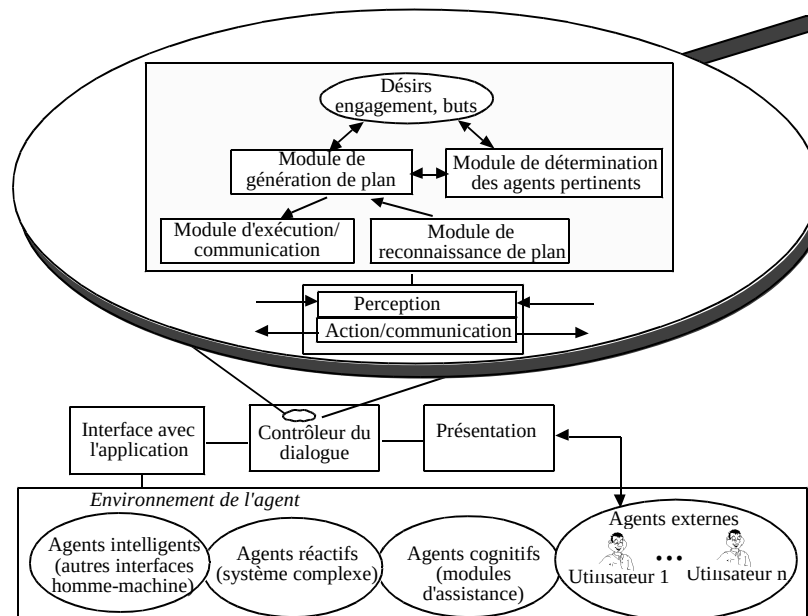


Figure 1.25 : Concepts d'agent intelligent dans le modèle de SEEHEIM



Dans ces conditions, l'interface homme-machine comprend un module capable de raisonner sur des connaissances relatives au désir et à l'engagement d'effectuer une action ou de répondre à un besoin, ainsi qu'aux différents buts à atteindre. Ce module s'engage à réaliser le(s) but(s) sélectionné(s). Si le but est réalisable, un plan d'intention est généré, sinon le but doit être sélectionné par un autre agent (par exemple si celui-ci entre dans le domaine de compétence d'un module d'assistance ou d'une autre interface homme-machine). Ce plan est constitué d'intentions d'engagement et de croyances sur l'univers qui l'entoure. Ce module intègre aussi un certain nombre d'actes tels que ceux de communication ou les actes probables des autres agents. Après l'élaboration d'un plan, l'agent l'exécute afin de réaliser le but pour lequel il s'est engagé. Ce but est réalisé par le module d'exécution/communication, dans la mesure où il peut comprendre une ou plusieurs actions, et/ou un ou plusieurs actes de communication vers un autre agent. Un module aura pour rôle de déterminer les agents pertinents susceptibles d'interagir avec l'agent actuel, pour atteindre le but.

Cette approche d'interface "intelligente" laisse présager de nombreux travaux et réflexions concernant : (i) le travail de l'utilisateur, qui aura à interagir de façon transparente avec un ensemble d'agents "intelligents", (ii) la conception des systèmes homme-machine qu'il s'agit de décomposer en agents, de même que (iii) les méthodes d'analyse et d'évaluation de ces systèmes. Enfin, la complexité d'une telle approche se retrouve dans la formalisation des différentes connaissances, particulièrement quand le nombre d'agents augmente. Cette approche est à l'état de prospective actuellement. Notons que certains des principes cités se retrouvent dans ARCHON, une architecture d'agents coopérants dédiée à l'assistance "intelligente" dans les systèmes complexes (AVOURIS et VAN LIEDEKERKE, 1993). COUTAZ et NIGAY (1991) se situent dans la même lignée.

#### ***1.3.2.6. Conclusion sur les architectures d'interfaces dites "intelligentes"***

L'évolution technologique des interfaces homme-machine a fait un bond considérable au cours de ces dernières années, s'orientant maintenant progressivement vers des concepts d'interfaces dites "intelligentes". La plupart des types d'interface présentés ont un objectif commun, qui est de s'adapter. Cependant, comme l'explique TENDJAOUI (1992), on retrouve pour chacun d'eux, un mode d'adaptation particulier ; par exemple, l'interface flexible s'adapte au système utilisé et aux préférences des utilisateurs, l'interface tolérante aux erreurs humaines s'adapte aux actions de l'utilisateur, l'interface adaptative s'adapte aux traitements cognitifs de l'utilisateur, etc.

Notre contribution à ce domaine réside dans le développement d'une interface dite "intelligente" pour le domaine de la supervision de système industriel complexe. Elle a été appelée "Module Décisionnel d'Imagerie" puisqu'elle se ramène à un module capable de décider du contenu des images à présenter. Ainsi, nous proposons d'utiliser un système à base de connaissance pour assurer les interactions entre le système, l'utilisateur et différents modules d'assistance. C'est une interface de type adaptative. Cependant, si nous lui associons des fonctionnalités d'assistance, elle sera très proche du système de dialogue de type "Opérateur assistant". *Nous la décrirons dans le chapitre suivant.*

### **1.3.3. Conclusion sur les architectures d'interface**

Dans cette section, nous avons proposé un bref état de l'art des modèles d'architecture d'interface homme-machine. Dans ce but, des approches s'intéressant aux aspects purement logiciels de l'interface ont d'abord été présentées, puis des concepts d'interfaces qualifiées à tort ou à raison "d'intelligentes" ont été positionnées par rapport au modèle de SEEHEIM. Au vu des différentes architectures expliquées, il est clair que le développement d'interface nécessite une approche pluridisciplinaire de par la variété des aspects, techniques ou humains, qu'il s'agit de considérer.

Bien entendu, en fonction du domaine dans lequel on se place (EAO, bureautique, systèmes grand public, supervision de système complexe...), nous avons déjà souligné que le développement de l'interface se basera sur une problématique, des contraintes, des méthodes, des outils et des modèles différents. Pour les applications visées dans ce mémoire, il devient encore plus évident que les compétences des développeurs doivent maintenant se situer bien au delà de la programmation ou de la configuration de logiciels graphiques. A ce sujet, les moyens de spécification, réalisation et évaluation mis à la disposition de l'équipe de développement font l'objet de la partie suivante.

## 1.4. SPÉCIFICATION, REALISATION ET EVALUATION DES INTERFACES HOMME-MACHINE

Grâce à l'apparition d'outils graphiques de plus en plus performants et à l'importance grandissante accordée aux démarches de prototypage, la tendance consiste actuellement à combiner les étapes de spécification, réalisation et évaluation des interfaces. C'est pourquoi ces trois dernières ont été rassemblées dans cette quatrième partie.

### 1.4.1. Spécification des interfaces homme-machine

La spécification des interfaces homme-machine doit être réalisée en collaboration par les différents intervenants (informaticien(s), utilisateur(s), spécialiste(s) des facteurs humains, expert(s) du système). A partir de la modélisation des tâches humaines et des différents utilisateurs, il s'agit de recenser rigoureusement les besoins, ergonomiques et techniques, puis de définir le nombre d'écrans à utiliser, l'architecture de l'interface homme-machine, l'enchaînement des vues, les modes de présentation des informations, les modes d'activation des différents outils d'aide, les modalités de dialogue homme-machine, etc. A ce sujet, des techniques issues du génie logiciel seront d'abord rappelées. Elles peuvent s'appuyer sur des recommandations ergonomiques, des guides de style d'interface et des normes ou des standards internes en vigueur dans le domaine d'application, points sur lesquels nous nous attarderons par la suite, figure 1.26.

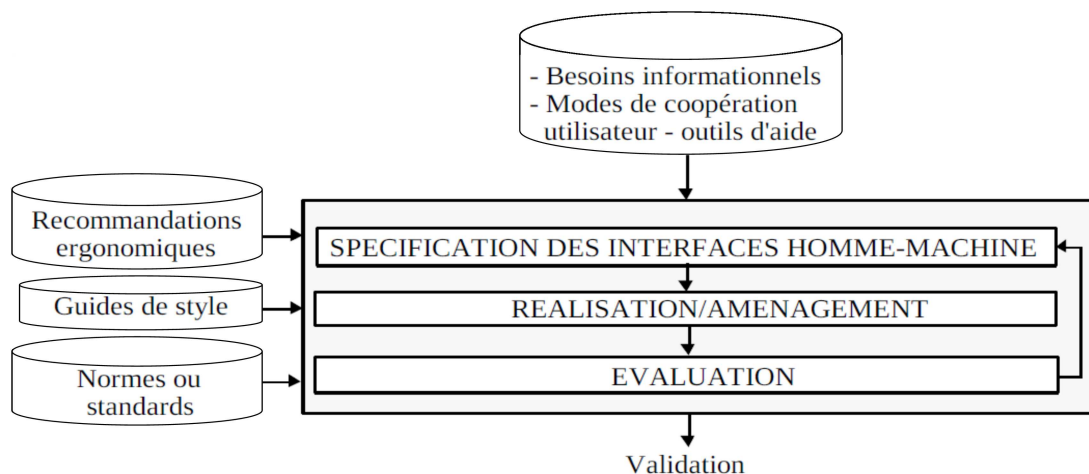


Figure 1.26 : Enchaînement de la spécification, la réalisation et et l'évaluation

#### 1.4.1.1. Techniques de spécification issues du génie logiciel

Pour des systèmes complexes, où les interactions homme-machine sont nombreuses, la spécification exige un travail particulièrement ardu. Pour spécifier les pages-écrans, la méthode certainement la plus employée consiste à dessiner sur papier les vues d'écran envisagées. Cette méthode n'est bien sûr pas suffisante, mais elle a l'avantage d'être pratique et de faciliter le dialogue entre les différents intervenants.

La spécification peut être facilitée par l'utilisation et/ou la combinaison d'un ensemble de techniques recensées par plusieurs auteurs tels COUTAZ (1990), VOEGTLIN (1991), EL MRABET (1991), HAMMOUCHE (1993) ou POULAIN (1994) :

- *Les techniques basées sur les langages formels et d'événements* : Celles-ci permettent de spécifier la syntaxe du dialogue homme-machine, sous la forme de séquences d'actions autorisées pour les utilisateurs. Les techniques principales sont : (i) les automates d'état fini caractérisés statiquement par un ensemble d'états et dynamiquement par un ensemble d'arcs étiquetés formalisant le passage d'un état à un autre, (ii) les grammaires indépendantes du contexte, où la spécification de l'interface peut se baser sur un formalisme de type BNF (Backus Naur Form), voir à ce sujet DANG (1988) (iii) les réseaux de Pétri ajoutant aux concepts d'état et de transition présents dans les automates d'état fini la notion d'activités parallèles (SOENEN et al., 1988 ; WILLEM et BILJON, 1988 ; BENAÏSSA, 1993) et (iv) les langages d'événements voyant une application interactive comme étant composée de plusieurs modules dirigés par des événements générés par l'utilisateur par exemple.
- *Les techniques basées sur les langages particuliers* : on distingue (i) les réseaux de menus (ou de pages-écrans), tel HYPERCARD, où l'utilisateur active différents menus, l'appel d'un menu pouvant entraîner l'apparition d'un autre, (ii) *les langages déclaratifs* permettant de spécifier des interfaces du type formulaire, où l'utilisateur fait de la saisie de texte dans des champs, ou sélectionne des options avec des menus ou des boutons, tels les systèmes COUSIN (HAYES et SZEKELY, 1983) et OPEN/DIALOGUE (SCHULER et al., 1985).
- *Les techniques basées sur l'utilisation des approches orientées objets* : la spécification de l'interface par une approche orientée objets offre des avantages indéniables en terme de génie logiciel. En effet, cette technique repose sur les notions de classe et d'héritage, facilitant la maîtrise de la complexité, l'extensibilité et la réutilisabilité des objets pour la construction des interfaces, et assurant la cohérence et l'homogénéité dans les modes de présentation et les dialogues (STROUSTRUP, 1988 ; FERBER, 1990 ; BOOCH, 1994). On retrouve cette approche dans de plus en plus de logiciels destinés aux applications de supervision.
- *Les techniques basées sur des spécifications graphiques et interactives* : le principe consiste à disposer sur l'écran, à l'aide de la souris, les différents composants de l'interface. Cette technique présente l'intérêt d'être très facile d'usage et permet à des non-informaticiens spécialistes de la conception d'interface de créer eux-mêmes les interfaces homme-machine, selon le principe suivi par MASAI (ILOG, 1989b) par exemple. D'après POULAIN (1994), de telles techniques ne peuvent être possibles que si elles sont associées à des outils informatiques performants, assurant la génération du code associé à l'interface. De plus, elles permettent d'accorder plus de temps à une conception ergonomique des interfaces, dans la mesure où les tâches de programmation sont réduites. Dans ce cas on dispose de plus de temps pour évaluer les interfaces produites ou encore pour proposer par exemple de nouveaux modes de représentation adéquats.

*Nous avons contribué à ce domaine sous la forme de l'étude d'un outil de prototypage intégré dans un atelier de génie logiciel permettant de produire des synoptiques industriels. Cet outil, utilisant ce principe de spécifications graphiques, sera décrit dans le chapitre deux.*

- Les techniques basées sur la génération automatique de spécification : elles visent, pour un type d'application bien cerné, à générer automatiquement des spécifications d'interface, à partir d'une description de l'application en question. A titre d'exemple, citons CONTROL PANEL INTERFACE (FISHER et JOY, 1987), IDL (FOLEY, 1987), SIROCO (NORMAND, 1992), ou encore les travaux de TARBY (1993). Il importe par la suite que ces spécifications soient facilement exploitables par le développeur, qui pourra générer l'interface en se basant sur les spécifications proposées, et la retoucher au besoin.

*Dans le cadre de nos recherches, nous avons contribué à ce domaine en mettant au point un tel principe de spécification intégré dans un système de conception ergonomique d'interface homme-machine de type supervision de système industriel complexe appelé ERGO-CONCEPTOR. Ce système original sera également décrit dans le second chapitre.*

Chaque technique citée, et bien entendu combinée éventuellement avec d'autres, peut faciliter le travail de spécification de l'interface homme-machine. Il est à ce sujet important de noter qu'elles se retrouvent progressivement dans des UIMS (User Interface Management System). De tels outils graphiques facilitent le prototypage d'applications interactives.

#### **1.4.1.2. Spécifications provenant de recommandations ergonomiques et de guides de style**

La spécification de l'interface peut être améliorée par l'utilisation de deux types de documents en rapport avec *l'ergonomie des logiciels* : les manuels ergonomiques et les guides de style.

- Les manuels ergonomiques consistent en une compilation de recommandations, guides, listes de contrôle, standards ergonomiques. Le plus connu est certainement celui de SMITH et MOSIER (1986), comprenant près de 1000 (!) règles d'ergonomie des logiciels. Celui-ci a inspiré de nombreux autres guides, utilisables pour la conception et/ou l'évaluation des interfaces homme-machine, intentionnellement moins exhaustifs et plus abordables que celui de SMITH et MOSIER. Citons les plus connus : SCAPIN (1986), SCHNEIDERMAN (1987), BROWN (1988) ou RAVDEN et JOHNSON (1989). Notons celui de GILMORE et al. (1989) dédié à la supervision d'applications industrielles complexes. Malgré leur grand intérêt, ces manuels présentent plusieurs inconvénients (SCAPIN, 1990 ; BASTIEN, 1991) : difficulté d'utilisation par des concepteurs ou évaluateurs non spécialistes des facteurs humains, manque d'information sur la façon de les utiliser, contradictions, etc.

*Lors de ces dernières années, nous avons eu l'occasion de rédiger plusieurs manuels lors de collaborations industrielles de recherche (Cf. chapitre suivant).*

- Les guides de style accompagnent généralement les boîtes à outils graphiques. Ils fournissent des recommandations concernant l'utilisation des objets de présentation qui sont disponibles (tels que des boutons, fenêtres, menus...), et visent à ce que ceux-ci soient utilisés de manière cohérente d'une application à l'autre. Ils sont illustrés de nombreux exemples graphiques au contraire de certains des manuels précédents. Des guides de style sont fournis avec la Tool Box du Macintosh, avec les boîtes à outils de Windows et Presentation Manager du monde IBM-Microsoft, avec celles de Motif et Open Look du monde Unix, etc. Toutefois, ces guides de style ne concernent d'aucune manière les aspects conceptuels, sémantiques ou méthodologiques liés au développement d'interface.

En s'inspirant de POULAIN (1994), il est utile de positionner ces deux types de documents par rapport à la manière dont les facteurs humains sont pris en compte, vis-à-vis de deux pratiques ergonomiques possibles : une pratique dite normative et une autre visant la conception (figure 1.27) :

- *L'ergonomie normative* correspond à une intégration de critères ergonomiques par le biais de manuels et/ou de guides de style, ce qui soulève les problèmes cités précédemment à savoir la limitation due à leur applicabilité. Les seuls transferts de connaissances humaines se font alors le plus souvent entre les experts et les développeurs uniquement, et cela dans le but d'analyser le fonctionnement de l'installation et d'exprimer les besoins en terme technique essentiellement. Dans

certains des cas seulement, le développeur, s'il est suffisamment sensibilisé aux facteurs humains, fait intervenir les utilisateurs dans son analyse.

- *L'ergonomie de conception* est possible par une intégration directe du spécialiste des facteurs humains dans l'équipe de développement. Son rôle est d'assister l'équipe de développement lors de la spécification de l'application, en regard des résultats d'une analyse des tâches prescrites faite auprès des experts de l'installation et des tâches réelles des utilisateurs. Il a également la maîtrise du contenu des manuels ergonomiques et des guides de style. Ce cas de figure est bien entendu idéal et est préconisé par la plupart des auteurs.

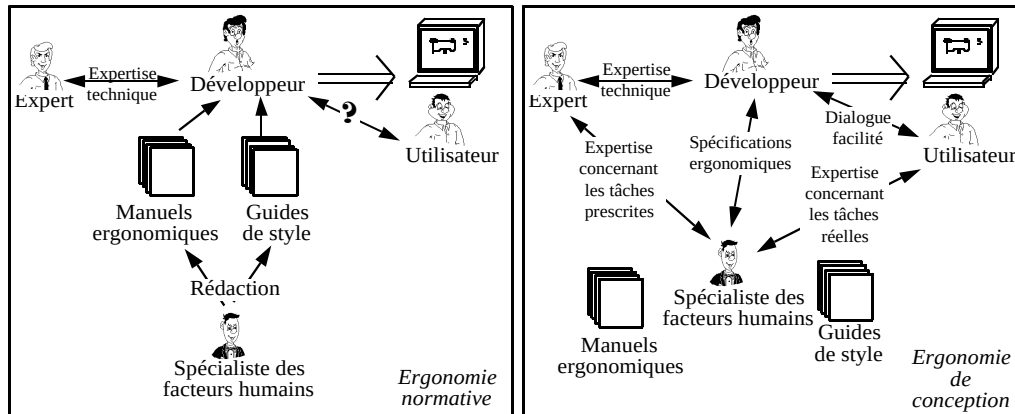


Figure 1.27 : Cas d'utilisation de manuels ergonomiques et des guides de style

#### 1.4.1.3. Spécifications provenant de normes ou des standards internes en vigueur dans le domaine d'application

La spécification des interfaces homme-machine doit rester cohérente avec les normes graphiques en vigueur dans le domaine d'application, si elles existent, ainsi que sur les éventuels standards graphiques internes à l'entreprise. On trouve par exemple des normes graphiques en sécurité routière, pour l'aide au déplacement dans les lieux publics, pour la description des symboles graphiques représentant des installations industrielles (voir ci-après), etc.

Depuis les années 70, des organismes de normalisation se sont penchés sur la normalisation des symboles graphiques et de leurs codes d'utilisation pour les affichages des composants représentant des installations industrielles : norme internationale ISO 3511, norme Britannique BS 1646, normes allemandes DIN 19 227 et DIN 19 228, et spécification ISA-S5.1-1984 (Revue MESURES, 1990).

En France, plusieurs industriels ont créé en 1986 une commission de travail chargée de porter sur le plan international un travail de réflexion sur les incohérences entre les normes existantes. Ce travail a conduit à la définition de la norme AFNOR E 04-203. A titre d'exemple, la figure 1.28 expose les symboles graphiques proposés par la norme pour schématiser un dispositif réglant. La forme 1 reprend le dessin de la norme ISA (surtout utilisée dans le secteur pétro-chimique) et la forme 2 reprend celui de la norme ISO (utilisée plus communément).

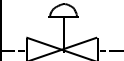



Symbole	
Forme 1	Forme 2
	
	

Figure 1.28 : Exemple de symbole graphique normalisé

#### 1.4.1.4. Conclusion sur la spécification des interfaces homme-machine

La spécification des interfaces homme-machine est un travail difficile qui demande une collaboration étroite entre les différents intervenants du projet. L'expression des spécifications peut être facilitée et accélérée par l'utilisation de nombreuses techniques issues du génie logiciel.

De plus, afin d'améliorer l'ergonomie du logiciel, il est possible de s'appuyer sur des manuels de recommandation, ainsi que sur des guides de style. Toutefois, l'utilisation de ces derniers sera confortée avantageusement par la présence d'un spécialiste de la communication homme-machine. Notons que la spécification devra également respecter les normes et/ou les standards du domaine d'application. Il en existe pour la supervision de systèmes industriels complexes.

L'équipe de développement a ensuite besoin d'outils graphiques puissants permettant de réaliser rapidement et d'animer les vues, afin d'en faciliter l'évaluation avec les utilisateurs.

### 1.4.2. Outils graphiques de réalisation des interfaces homme-machine

Le marché informatique actuel offre un éventail varié d'environnements graphiques facilitant la réalisation et la modification d'interface homme-machine en réduisant l'écriture de code et en permettant la réutilisabilité des programmes. Ces environnements graphiques ont été recensés et étudiés par plusieurs auteurs tels PERLMAN (1988), JURAIN (1991), MEINADIER (1991) et surtout COUTAZ (1988 ; 1990 ; 1991), qui ont fait abstraction des différents domaines d'application.

Il nous paraît donc utile d'axer notre discussion vis-à-vis du domaine qui nous préoccupe dans ce mémoire, en l'occurrence la supervision nécessitant l'animation de courbes, historiques, symboles ou barre-graphes en temps réel. Dans cette optique, il est possible de classer grossièrement les environnements graphiques utilisables en supervision selon plusieurs catégories en nous inspirant des études de COUTAZ. Les trois paragraphes suivants discutent sommairement de l'utilisation de boîtes à outils, d'éditeurs d'interfaces et de progiciels spécialisés.

#### 1.4.2.1. Utilisation de boîtes à outils

La première catégorie est basée sur l'utilisation de boîtes à outils, et s'adresse à des programmeurs. Elles facilitent l'écriture de programmes de dialogue à partir des différents dispositifs d'entrées/sorties d'information (écran, souris, clavier, etc).

Par exemple, dans le monde UNIX et son standard de fait X11, X-Toolkit (MIT, 1987) est certainement la boîte à outils la plus connue (si nous omettons les standards Motif et Open Look destinés aux interfaces de dialogue). Située au-dessus de la bibliothèque graphique Xlib, elle offre un ensemble d'objets prédéfinis appelés Widgets. Toutefois, cette boîte ne fournit pas dans sa version de base d'objets

graphiques habituellement utilisés en supervision. Cette remarque est valable également pour la boîte à outil AIDA (ILOG, 1989a) associée au générateur MASAI (Cf. 1.4.2.2), de plus en plus répandue en France. Par contre DV-Tools de l'environnement DataViews (V.I. CORPORATION, 1988) offre toutes les fonctionnalités nécessaires pour l'animation d'images de supervision.

Les boîtes à outils exigent un apprentissage souvent long pour le programmeur qui doit de surcroît complètement concevoir les enchaînements entre vues. Cependant, elles ont l'avantage d'intégrer des critères ergonomiques de base, en particulier de standardisation de la présentation de l'information (menus prédéfinis, format et manipulation de fenêtres, présentation d'icônes par exemple) ou encore de contrôle des entrées de l'utilisateur grâce à la sélection d'information par souris. Elle sont également caractérisées par la portabilité des logiciels interactifs résultants et l'extensibilité des fonctions qu'elles intègrent. Toutefois, le manque de souplesse qu'implique leur utilisation ne facilite pas le prototypage.

#### ***1.4.2.2. Utilisation d'éditeurs d'interfaces***

Nous nous intéressons ici uniquement aux éditeurs d'interfaces permettant de spécifier *interactivement* l'interface. Ceux-ci peuvent s'adresser à des développeurs non-informaticiens, et consistent en des outils graphiques interactifs facilitant le prototypage, tels DV-Draw (V.I. CORPORATION, 1988) associé à DV-Tools, SL-GMS (SHERRILL-LUBINSKI CORPORATION, 1990) ou MASAI (ILOG, 1989b) associé à AIDA. Il faut toutefois souligner que ces éditeurs exigent le plus souvent de programmer certaines parties du dialogue homme-machine.

Un énorme avantage que possèdent les générateurs d'interfaces actuels est d'être facilement exploitables par des spécialistes de l'ergonomie du logiciel et de produire des interfaces immédiatement évaluables. Cependant, ils sont parfois limités par les modes de représentation disponibles (même remarque que pour certaines boîtes à outils). Même si le développeur a accès avec certains éditeurs tel DV-Draw à des modes de représentation classiques en supervision tels des barre-graphes, des courbes, des symboles, etc, il peut éprouver les pires difficultés à parvenir à une représentation d'information moins conventionnelle. Afin de pallier ce problème, certains environnements offrent la possibilité de programmer de nouveaux modes de représentation.

La plupart des éditeurs graphiques du marché ne sont pas prévus au départ pour la supervision de systèmes complexes. Ils s'intègrent donc dans des environnements logiciels plus ou moins adaptés aux contraintes temporelles sévères liées à certaines applications. Dans ce cas, il peut être nécessaire de remettre en cause l'utilisation d'un éditeur et de s'orienter sur une boîte à outils, plus "proche de la machine", ou un progiciel spécialisé.

#### ***1.4.2.3. Utilisation de progiciels spécialisés***

La troisième catégorie regroupe les progiciels graphiques spécialisés dans la supervision temps réel de systèmes industriels complexes. Ceux-ci sont généralement interfaçables directement avec plusieurs automates et calculateurs du marché. Ces progiciels consistent en des squelettes d'applications interactifs et configurables, et regroupent le code réalisant les fonctions usuelles de l'interface homme-machine sous la forme d'un logiciel réutilisable et complétable selon les besoins.

La démarche consiste alors à configurer l'outil par rapport à l'application. Pour ce faire, on dispose d'outils de déclaration des caractéristiques de l'installation, de création de synoptiques, mais aussi de configuration de modules d'assistance : affichage d'alarmes, de conseils d'action, de valeurs prédites, etc. Ainsi, au contraire des deux autres catégories d'outils citées, la plupart des progiciels graphiques spécialisés intègrent des fonctionnalités de base d'assistance à l'utilisateur lors de la réalisation de ses tâches. On recense actuellement une centaine de progiciels disponibles en France (PRADENC, 1992).

#### 1.4.2.4. Conclusion sur les outils graphiques de réalisation

Il existe actuellement de nombreux produits permettant la réalisation des interfaces homme-machine. Cependant, ils ne sont pas tous adaptés aux contraintes des applications liées à la supervision de certains systèmes industriels complexes. Ces contraintes concernent en particulier les temps de réponse et la variété des modes de présentation de l'information.

De plus, la facilité plus ou moins grande de ces produits pour construire les interfaces homme-machine et leur apporter des modifications aura une incidence sur la volonté des développeurs à les évaluer et à prendre en compte les remarques des utilisateurs. A ce sujet, des méthodes d'évaluation font l'objet de la partie suivante.

#### 1.4.3. Méthodes utilisables pour l'évaluation des interfaces homme-machine

Dans la plupart des projets de développement d'interfaces homme-machine, l'évaluation de leur utilité et de leur utilisabilité est souvent bâclée, maladroitement menée, voire ignorée. Dans ce cas, l'interface peut ne pas répondre aux besoins des utilisateurs. Pourtant, il existe actuellement de nombreuses méthodes et modèles issus de l'ingénierie et des sciences cognitives permettant d'évaluer la qualité des interfaces homme-machine.<sup>9</sup>

SENACH (1990a ; 1990b) a classé ces méthodes et modèles selon deux approches : (i) *l'approche empirique* se base sur la mesure des performances d'utilisateurs représentatifs de la population finale, lors d'expérimentations, alors que (ii) *l'approche analytique* prend en compte des modèles de l'interface et de l'interaction homme-machine, l'évaluation étant réalisée en comparant l'interface à un modèle de référence. Cette section résume le chapitre 7 de l'ouvrage de KOLSKI (1993) et met à jour la classification de SENACH. Pour chaque méthode (ou classe de méthode), son principe de base est résumé et discuté brièvement par rapport au domaine d'application visé.<sup>10</sup>

##### 1.4.3.1. L'approche empirique

L'approche empirique permet d'évaluer l'ergonomie de l'interface à partir du recueil et de l'analyse de données provenant de son utilisation par des utilisateurs représentatifs de la population finale, et ceci dans un environnement d'évaluation le plus proche possible de celui d'utilisation. Deux approches sont distinguées : (i) les tests de conception, utilisable lorsqu'il n'existe pas d'expérience d'utilisation du système, et (ii) le diagnostic d'usage, dans le cas contraire, figure 1.29.

<sup>9</sup> Nous focalisant sur les méthodes et modèles permettant d'évaluer la qualité ergonomique des interfaces, nous avons délibérément omis de discuter de la notion de qualimétrie, même si l'aspect ergonomique pointe timidement le bout du nez dans les méthodes de qualimétrie actuelles (voir par exemple l'ouvrage de FORSE, 1989).

<sup>10</sup> Il existe d'autres classifications des méthodes d'évaluation dans la littérature, le lecteur consultera par exemple BALBO et COUTAZ (1992), SWEENEY et al. (1993) ou WILSON (1990).



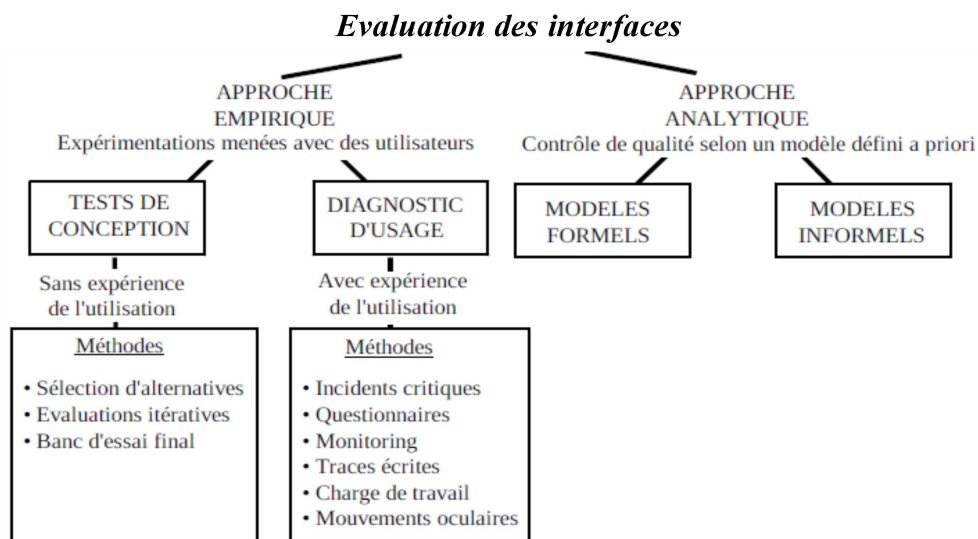


Figure 1.29 : Approche empirique d'évaluation de l'interface homme-machine

#### 1.4.3.1.1. Les tests de conception

Ce type d'évaluation peut être mis en oeuvre lorsqu'il n'existe pas encore d'expérience d'utilisation du système. Des tests sont alors réalisés tout au long du processus de conception. Trois démarches seront successivement décrites (figure 1.30).

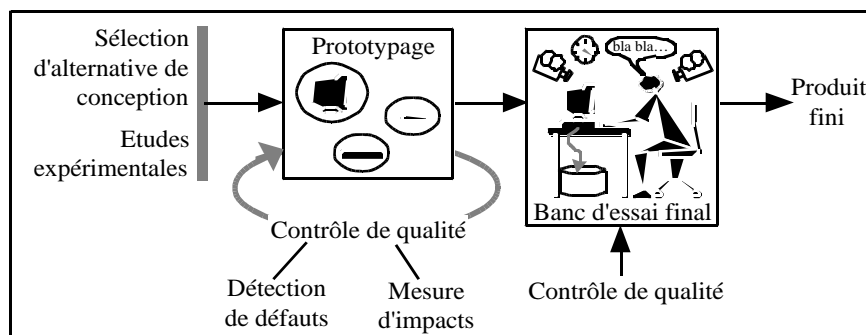


Figure 1.30 : Tests de conception

La méthode des *tests de conception par sélection d'alternatives* est utilisable lorsqu'il n'existe pas de critère de choix évident entre plusieurs possibilités, et le recueil des données empiriques doit alors permettre de hiérarchiser les solutions envisagées au départ. Cette méthode est la plupart du temps conduite en amont du prototypage (voir ci-dessous). Des alternatives sont donc évaluées par rapport aux objectifs du système final, les résultats servant de base pour les étapes suivantes de la démarche de conception. Une des limitations à ce type d'évaluation est qu'elle est effectuée souvent en dehors du contexte réel de la tâche, de ses contraintes, de son environnement. En effet, une alternative obtenant de médiocres résultats en phase d'évaluation peut s'avérer meilleure dans une situation de travail, en se rattachant à d'autres informations suivant la logique de l'utilisateur pour la tâche courante. Cette technique peut être utilisée avantageusement dans un contexte de supervision où la complexité des tâches et la multiplicité des supports d'information utilisés entraînent rapidement la mise en évidence de nombreuses alternatives.

Une autre méthode concerne *l'évaluation itérative de l'interface* (discutée dans la partie 1.1.1.3. suite à la présentation du modèle spirale). Dans la section 1.4.2 était clairement mise en évidence l'apparition actuellement sur le marché de nombreux environnements graphiques facilitant la réalisation et la modification d'interface homme-machine par la réduction de l'écriture de code et la réutilisabilité des

programmes. La plupart de ces outils facilitent la réalisation itérative de plusieurs versions plus ou moins avancées de l'interface, qu'il est donc possible d'évaluer au fur et à mesure du projet. Le prototypage a pour objectif de minimiser les coûts de développement de l'interface tout en visant l'optimisation progressive de sa qualité selon des critères ergonomiques généraux, ainsi que des critères spécifiques à l'application. L'efficacité de cette démarche dépend directement des possibilités de *réutilisation* des composants logiciels et de *réaménagement* rapides en fonction des résultats de l'évaluation. La dernière version du prototype est celle considérée comme satisfaisante vis-à-vis des besoins des utilisateurs pour les différentes tâches à réaliser. La rentabilité du prototypage est totale si cette dernière version peut être utilisée en tant que produit final. Le prototypage rend nécessaire la participation des utilisateurs à la démarche de conception, tout en servant de support de dialogue entre ces derniers et les concepteurs.

*Nous avons contribué au prototypage lors d'un projet industriel ayant pour objectif de réaliser un atelier facilitant d'une part la spécification de synoptiques de supervision et d'autre part leur réalisation et évaluation. Cette contribution sera présentée plus loin.*

Une autre méthode consiste à mettre en place *un banc d'essai final*. A ce sujet, un système doit respecter de nombreux critères avant de pouvoir être considéré comme *utilisable*. (GOULD, 1988 ; NIELSEN, 1993). Avant toute commercialisation, il importe d'évaluer globalement chacun de ces composants et d'en faire ressortir les points forts et les points faibles. Dans ce but, des *stations d'évaluation*, faisant office de *banc d'essai final*, font leur apparition. Il en existe par exemple dans les locaux des constructeurs de logiciels LOTUS, MICROSOFT ou WORDPERFECT. Spécialement aménagés, elles permettent le recueil empirique de données objectives et subjectives à l'aide de méthodes d'observation élaborées (LUND, 1985 ; WILSON et CORLETT, 1990). Dans le domaine du supervision de système complexe, les stations d'évaluation prennent la forme de simulateurs réalistes (LEQUESNE, 1991). La simulation permet aux utilisateurs de se familiariser avec le système, de l'évaluer dans des conditions proches de l'exploitation tout en étant plus économiques, et de proposer des améliorations visant à résoudre les difficultés liées à l'accomplissement de certaines tâches. Bien entendu, la simulation entraîne la nécessité de développer une plate-forme spécifique où l'environnement dynamique de l'application cible doit être reconstitué.

#### 1.4.3.1.2. Le diagnostic d'usage

Cette évaluation est effectuée lorsqu'il existe une expérience d'utilisation du système dans sa globalité. Six techniques, présentées successivement, permettent de diagnostiquer des fonctions ou des modes de représentation défaillants, inutiles, difficiles à exploiter, etc.

*La méthode des incidents critiques* a été utilisée pour la première fois par FLANAGAN (1954, cité par SINCLAIR, 1990) pour étudier des accidents d'avions évités de justesse. Elle consiste à recueillir systématiquement les dysfonctionnements du système homme-machine à partir d'entretiens avec les utilisateurs et d'observations effectuées sur leur poste de travail. Chaque incident est décrit sous la forme de court récit. Les récits font l'objet d'une classification regroupant des incidents instances d'un même problème. Les problèmes sont ensuite regroupés en classes plus générales. Elles peuvent être ensuite utilisées pour définir les premières fonctionnalités d'un nouveau système, réaliser des aménagements, préciser les objectifs de formation, etc. Cette méthode permet rapidement d'effectuer un diagnostic global des principaux dysfonctionnements. Elle est employée dans la plupart des situations d'analyse et d'évaluation, prenant éventuellement la forme de variantes.

*Le questionnaire d'utilisation* permet à l'évaluateur de recueillir des informations subjectives sous une forme sûre et structurée, propice à l'analyse. Elle a l'avantage de permettre à l'utilisateur de travailler "à tête reposée", même si ceci est susceptible d'introduire un biais dans ses réponses puisqu'il n'est pas toujours en mesure de se souvenir des difficultés rencontrées lors des différentes situations de travail. Plusieurs travaux ont été consacrés à la manière de réaliser un questionnaire (voir GRAY, 1975 ou

SINCLAIR, 1990). En supervision de système complexe, la réalisation d'un questionnaire peut exploiter des travaux spécifiques à ce domaine : GERTMAN et al. (1982) ont mis au point un questionnaire, dont on trouvera un exemple d'utilisation dans COCKERILL-SAMBRE (1984). Il a été ensuite largement étendu par les auteurs (voir GILMORE et al., 1989).

*Le monitoring*, ou "mouchard électronique", permet de recueillir automatiquement des données objectives à l'aide de l'ordinateur. Cette méthode peut renseigner l'évaluateur sur les activités et les performances d'un utilisateur, que ce soit en situation réelle ou en simulation. Elle est généralement associée à des questionnaires et des verbalisations. Elle n'est pas intrusive, ne perturbant pas la réalisation des tâches, et n'apportant donc pas de biais dans les données recueillies. Celles-ci sont généralement traitées automatiquement afin d'obtenir des informations variées : stratégies utilisées, séquencement réel des tâches, performance obtenue, identification des erreurs d'utilisation, etc (MAGUIRE et SWEENEY, 1989 ; DRURY, 1990). Par les nombreuses données temporelles fournies, cette technique peut contribuer à l'amélioration des interfaces et des modules d'assistance.

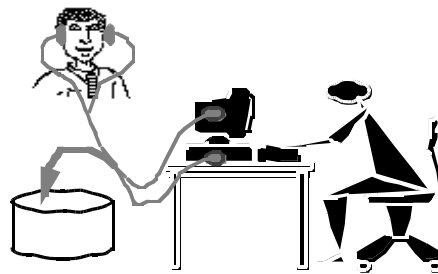


Figure 1.31 : Principe du mouchard électronique

*L'analyse des traces écrites* par l'utilisateur lors du travail est susceptible de faciliter la mise en évidence de problèmes d'utilisation. Dans les salles de supervision, on retrouve le plus souvent des rapports qui décrivent les événements survenus, les défauts, les arrêts, les redémarrages, etc. Certaines d'entre elles peuvent mettre en lumière des lacunes dans l'interface. Dans certaines salles, on retrouve aussi des cahiers de doléances, surtout dans les premiers mois faisant suite au lancement d'une nouvelle installation ou imagerie. Ces cahiers peuvent être d'un apport précieux pour l'amélioration de l'interface, en servant d'outil de dialogue entre utilisateurs et développeurs.

*La charge de travail* est une grandeur utile pour l'évaluation des interfaces homme-machine. En raison de la complexité des tâches à effectuer par les utilisateurs dans le domaine de la supervision, il est particulièrement instructif d'évaluer dans quelles mesure les outils graphiques influent sur leur charge de travail. Il existe de nombreuses méthodes d'estimation de la charge. Celles-ci peuvent être soit subjectives, consistant à demander à l'utilisateur d'indiquer a posteriori son sentiment de charge, le plus souvent au moyen d'échelles subjectives, soit objectives, utilisant des paramètres physiques ou physiologiques mesurés pour estimer la charge de travail (MILLOT, 1988). Les résultats obtenus permettent de mettre en évidence des points sensibles pour les outils graphiques utilisés, et de procéder ensuite à des aménagements de ceux-ci en cas de difficultés lors de l'exécution de certaines tâches (BERGER, 1992 ; RIERA, 1993).

Une autre étude consiste en *l'analyse des mouvements oculaires*. Il s'avère que dans la supervision, il est fait énormément appel à la perception visuelle des informations. Dans ces conditions, la direction absolue du regard devient un élément utilisable en parallèle avec d'autres moyens pour analyser l'activité d'un utilisateur devant un écran ou un ensemble de supports d'informations. Il est possible, à partir de l'étude de son activité oculo-motrice, d'étudier la manière dont il recherche et localise les informations utiles en fonction des différentes situations, de mettre en évidence des stratégies utilisées et d'identifier des lacunes liées aux outils lors de certaines situations de crise. Il existe de nombreuses techniques de mesure des mouvements oculaires (voir ACIER, 1990 et DERICQUEBOURG, 1991). Bien que leur mise en place pose certains problèmes techniques ou ergonomiques, celles-ci paraissent désormais suffisamment

maîtrisées pour contribuer à l'évaluation des stratégies visuelles de l'utilisateur en salle de supervision, et donc à l'aménagement des interfaces (BERGER, 1992 ; SIMON, 1993).

#### 1.4.3.2. L'approche analytique

L'approche analytique vise à contrôler la qualité de l'interface selon un modèle défini *a priori* : ce modèle peut être soit formel, soit informel, figure 1.32.

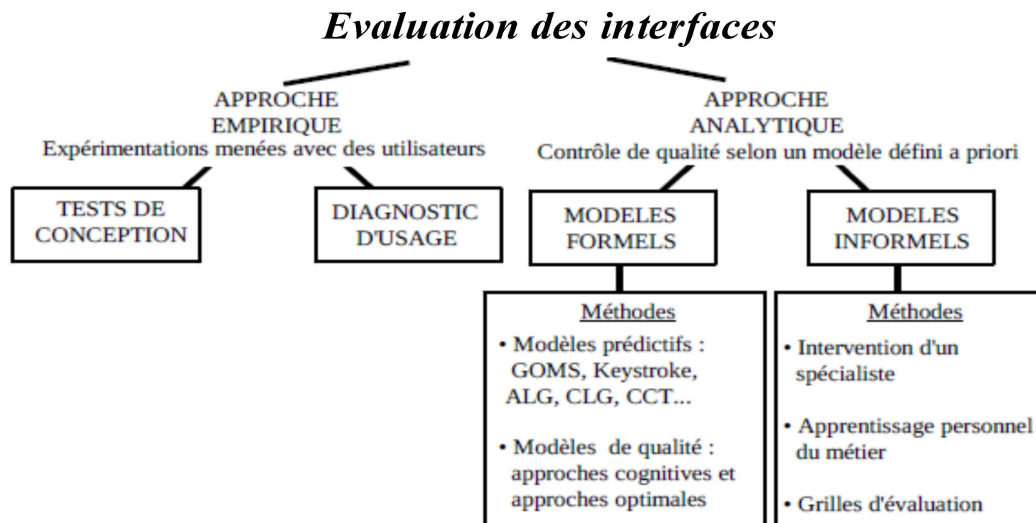


Figure 1.32 : Approche analytique d'évaluation de l'interface homme-machine

##### 1.4.3.2.1. Les modèles formels

Dans l'approche analytique, l'évaluation de l'interface homme-machine peut être assistée de démarches et d'outils qualifiés de *formels* : on distinguera alors les modèles prédictifs et les modèles de qualité de l'interface (figure 1.32).

Plusieurs *modèles formels dits prédictifs* sont progressivement mis au point, partant de l'hypothèse que certaines performances de l'utilisateur peuvent être prédites, et donc considérées lors de la conception de l'interface. Par exemple :

- *Le modèle GOMS* (Goal, Operator, Method, Selection), défini par CARD et al. (1983), permet de modéliser le comportement de l'utilisateur à différents niveaux d'abstraction, depuis la tâche jusqu'aux actions physiques en termes de buts, d'actions élémentaires, de méthodes pour réaliser un but et de règles de sélection de méthodes. Bien qu'orienté vers la conception, il peut servir pour l'évaluation. Les prédictions concernent la durée de réalisation des tâches, le choix des méthodes et le choix des opérateurs. La démarche consiste à décrire hiérarchiquement l'utilisation d'un dispositif, puis à prédire les performances, et enfin à valider celles-ci dans un contexte expérimental. Ce modèle est très réducteur puisqu'il ne permet de prédire les performances que d'un utilisateur expert ne commettant pas d'erreur.
- *Le modèle Keystroke*, issu également de CARD et al. (1983), est dérivé de GOMS. Il concerne les actions physiques d'un utilisateur expert lorsqu'il effectue une tâche. Il permet de prédire ses performances à partir de : (i) la description préalable de la tâche, (ii) du langage de commande du système, (iii) de paramètres caractéristiques des capacités motrices de l'utilisateur et des temps de réponse du système, (iv) et de la méthode de réalisation de la tâche. Il vise la prédiction du temps

d'exécution d'une tâche. En règle générale, il hérite des avantages et des limitations de GOMS. Ses principes possèdent toutefois l'intérêt d'être facilement compréhensibles et de fournir des valeurs numériques simples, bien que sa mise en oeuvre pose rapidement de nombreux problèmes.

- *Le modèle ALG* (Action Language Grammar), provenant de REISNER (1981 ; 1984), permet de représenter sous forme de grammaire les actions de l'utilisateur de l'interface, sous la forme de règles de production du type : *POUR effectuer-telle-action FAIRE telles-opérations*. ALG vise un double objectif : permettre la comparaison entre des alternatives de comparaison selon un critère de facilité d'utilisation, et faciliter l'identification des choix de conception susceptibles d'engendrer des erreurs d'utilisation. ALG permet donc de ramener en quelque sorte l'évaluation lors de la conception de l'interface. Cependant, ce langage ne prend pas en compte la dynamique et les types de modes d'entrées/sorties, ceux-ci ayant une influence indéniable sur la complexité des tâches en fonction du contexte opérationnel.
- Bien qu'orienté vers la conception, *le modèle CLG* (Command Language Grammar) proposé par MORAN (1981) permet comme le précédent une évaluation prédictive de la facilité d'utilisation de l'interface. Un système est décrit en CLG à différents niveaux d'abstraction (tâche, sémantique, syntaxique, interaction, matériel). Le modèle permet de localiser les décisions d'évaluation qui seront critiques pour l'utilisateur. Il propose d'ailleurs différentes métriques permettant de prédire l'efficacité de l'interface. CLG possède comme limitations l'absence d'aide à la structuration et l'absence de moyens d'expression du parallélisme et des interruptions, éléments indispensables de modélisation de l'accomplissement d'une tâche.

De manière générale, ces modèles formels demandent encore à être affinés pour être utilisés dans un contexte de supervision, dans la mesure où ils ont fait surtout l'objet de validations pour des tâches simples de bureautique. Toutefois, leurs principes devraient progressivement se retrouver dans les méthodes du génie logiciel.

Il existe d'autres modèles dits *modèles formels de qualité de l'interface*. Ceux-ci s'intéressent aux propriétés mesurables de l'interface, d'un point de vue plus technique que les modèles précédents. La classification de SENACH (1990a) distingue : (i) *l'approche cognitive de la qualité de l'interface* qui prend en compte les traitements effectués sur l'information, (ii) *l'approche optimale de sa qualité* qui évalue l'interface selon des critères quantitatifs liés à la présentation de l'information. *Deux de nos contributions seront positionnées dans les modèles formels de qualité.*

Dans les modèles formels de qualité de l'interface, on distingue des approches qualifiées de *cognitives*. Certaines s'intéressent à l'identification du modèle mental d'un utilisateur ayant à naviguer dans les menus, dans un but de meilleure spécification de ceux-ci (PAAP et ROSKE-HOFSTRAND, 1988). D'autres recherches visent la prédiction des *difficultés d'utilisation de l'interface*, en tentant d'aller plus loin dans l'utilisation des modèles mentaux. Par exemple, KIERAS et POLSON (1985) émettent à ce sujet l'hypothèse suivante : la complexité d'un dispositif dépend de l'écart entre les connaissances requises pour l'utiliser et le modèle mental élaboré par l'utilisateur. Plus celui-ci est proche du modèle "objectif" du système informatique, plus ce système est facile à apprendre et à utiliser. Mais, la difficulté consiste alors à identifier les métriques contrôlant les écarts entre les deux représentations (KELLOG et BREEN, 1983). D'autres recherches concernent l'évaluation de *la cohérence conceptuelle* de l'interface qui, au niveau de l'utilisation des fonctions et de la présentation d'information, a une énorme influence sur sa facilité d'utilisation et son apprentissage (KELLOG, 1987). Ces recherches sont très utiles puisqu'elles permettent de mieux comprendre le modèle cognitif que se forgent les utilisateurs de l'interface à partir de certaines de ses composantes caractéristiques.

Dans les modèles formels de qualité de l'interface, d'autres approches sont dites *optimales* :

- Certaines recherches se basent sur un *modèle de comportement de l'interface*. MONK et DIX (1987) partent de l'hypothèse qu'une interface peut être évaluée en fonction de quatre principes de base : (i) la prédictibilité de son comportement après chaque interruption, (ii) la simplicité d'usage, (iii) la cohérence au niveau de la sémantique des commandes, (iv) la réversibilité de chaque commande. Ils proposent de décrire un dispositif à l'aide de règles de type "Action-Effet", où chaque action est mise en relation avec les effets observables. Chacune des règles est ensuite analysée par rapport aux quatre principes de base. Cette approche oblige le concepteur à formuler et traiter certains problèmes que l'interface est susceptible de poser, mais elle possède des limitations liées au manque de relation entre les principes de base et des connaissances relatives aux tâches et aux utilisateurs.
- D'autres recherches s'intéressent à *la complexité perceptive des affichages*, et étudient la présentation de l'information, telle qu'elle est perçue par l'utilisateur. Elles peuvent se baser sur les nombreuses règles ergonomiques recensées dans ce domaine depuis une vingtaine d'années. Par exemple, TULLIS (1984, 1988) s'est orienté sur une démarche automatique d'évaluation d'écran, son programme étant capable de mesurer la densité globale d'information, la densité locale, le nombre de groupes d'informations distincts, la taille moyenne des groupes, le nombre d'items et la complexité de l'affichage. STREVELER et WASSERMAN (1984) ont suivi globalement la même démarche. Notons également les systèmes experts d'évaluation d'interfaces à base de menu en cours de réalisation par GIBBONS (1992) et LOWGREN et NORDQVIST (1992). Leur principale limitation est que l'évaluation s'effectue écran par écran, sans s'intéresser aux aspects sémantiques de l'application. Cependant, au vu des résultats obtenus, elles peuvent conduire à terme à des outils opérationnels pour l'ingénierie d'évaluation.

*Nous avons contribué à ce domaine par l'étude et la mise en œuvre d'un système original appelé SYNOP, décrit dans le chapitre 2.*

- D'autres recherches visent la *génération automatique d'affichage* respectant des concepts ergonomiques de base. Comme pour les trois systèmes précédents, on peut s'attendre à terme à des réalisations industrielles concrètes selon les principes cités. Par exemple, le système développé par MACKINLAY (1986) a pour objectif d'automatiser la présentation de l'information, indépendamment de l'application. Les différentes présentations graphiques correspondent à des phrases de langages graphiques s'apparentant à des langages formels. Le formalisme comporte un ensemble de primitives graphiques et d'opérateurs de composition, permettant la génération de modes de représentation de plus haut niveau. Le système développé par PERLMAN (1987) suit globalement la même démarche.

*Notre contribution à ce domaine concerne l'étude et la mise en œuvre d'un système appelé ERGO-CONCEPTOR décrit dans le chapitre suivant.*

#### 1.4.3.2.2. Les modèles informels

Le premier type d'évaluation consiste à *faire intervenir un spécialiste en communication homme-machine* pour juger de la qualité d'une interface et proposer des améliorations. Cette méthode est sans doute celle qui donne les meilleurs résultats. Une des critiques les plus courantes adressées toutefois à ce type de méthode est que chaque spécialiste se focalise sur des aspects particuliers de l'interface et fonde son évaluation sur une démarche globale qui lui est propre (HAMMOND et al., 1984 ; MOLICH et NIELSEN, 1990). L'idéal consisterait donc à faire intervenir plusieurs spécialistes, à confronter puis synthétiser leurs résultats (POLLIER, 1991). Malgré ces limitations, pour l'évaluation d'interfaces de supervision, cette méthode est à préconiser dans la mesure où elle permet de diagnostiquer rapidement un certain nombre d'erreurs de conception.

*L'apprentissage personnel du métier* consiste pour l'évaluateur à se mettre pour un temps à la place d'utilisateurs et à apprendre leur métier. SPERANDIO (1991) ne recommande pas cette méthode : elle est à exclure lorsque les tâches sont compliquées, elle est rarement utile, l'évaluateur a tendance à se prendre comme modèle, ne peut que devenir un utilisateur sans expérience, donc éprouver des difficultés à analyser objectivement le travail. Cependant, pour des tâches bien où l'évaluateur est en fait un utilisateur potentiel (par exemple en bureautique), la mise en oeuvre de la méthode entraîne moins de difficultés et en tout cas permettre au moins l'évaluation de la prise en main du système. Dans le domaine qui nous préoccupe, cette méthode pose un énorme problème : il est hors de question d'utiliser l'imagerie en ligne au risque de modifier des paramètres de régulateurs, des seuils d'alarmes ou des valeurs de consigne. Afin d'améliorer la compétence de dialogue avec les utilisateurs et de mieux appréhender l'utilisation de fonctionnalités de l'imagerie, il peut être toutefois possible pour l'évaluateur d'utiliser l'imagerie hors ligne, sans conséquence sur le système.

Des *grilles d'évaluation* ont pour objectif d'assister l'évaluateur en recensant des paramètres caractérisant l'ergonomie d'une interface. Pour chacun de ces paramètres, l'interface est notée systématiquement selon une échelle comportant plusieurs points. Par exemple, une méthode d'évaluation par "check-list" a été mise au point par RAVDEN et JOHNSON (1989) dans le cadre d'un projet ESPRIT. Elle est utilisable par des concepteurs, des spécialistes des facteurs humains et des utilisateurs finaux du système. Dans le domaine spécifique de la supervision, GILMORE et al. (1989) fournissent une liste très complète de questions facilitant l'évaluation. Malgré les progrès restant à accomplir dans la définition et l'optimisation de telles grilles, l'application de cette méthode peut d'ores et déjà s'appuyer sur des travaux ayant obtenu des résultats prometteurs. Elle peut être d'un apport très bénéfique pour l'évaluateur puisqu'elle l'oblige à ne négliger aucun des aspects de l'interface.

#### **1.4.3.3. Conclusion sur les méthodes d'évaluation**

Il existe une grande variété de méthodes contribuant à l'évaluation des interfaces homme-machine, et ceci, à l'aide particulièrement du prototypage et des modèles prédictifs, au niveau même de la conception. Cette constatation rejoint le constat fondamental de SENACH (1990a) : "cette organisation va à l'encontre des idées reçues en matière d'évaluation : elle montre que celle-ci constitue un processus parallèle à la conception, qu'elle ne peut pas être réduite à une simple étape ponctuelle et qu'une équipe d'évaluation doit pouvoir utiliser un ensemble d'outils allant des techniques cliniques aux méthodes formelles à chaque étape de la vie d'un produit. Elle ne pourra être effective que dans la mesure où des outils fiables, systématiques et faciles à utiliser seront disponibles". Dans le troisième chapitre, nous tenterons d'exploiter plus en profondeur ce constat en visant un positionnement de chaque méthode dans le cycle de développement.

## **CONCLUSION**

Ce chapitre était consacré à un état de l'art sur le domaine de recherche. En quatre parties, il a été possible de constater la richesse et la variété des approches, issues de l'ingénierie et des sciences cognitives. Pourtant, cet état de l'art n'était aucunement exhaustif, puisqu'il se focalisait sur les interfaces homme-machine utilisées dans des systèmes industriels complexes. Mais, en raison des critères économiques et sécuritaires liés le plus souvent à leur utilisation dans des situations parfois extrêmes, leur développement nécessite des approches spécifiques, et en tout cas pluridisciplinaires. Il s'avère donc que de nouvelles méthodes, connaissances, modèles et outils doivent maintenant de plus en plus être connues et utilisées. Ce chapitre a tenté d'en donner un aperçu.

La première partie s'est centrée sur les aspects méthodologiques du développement. Au vu des limites mises en évidence pour la plupart des modèles et méthodes issus du génie logiciel, il a été nécessaire de se définir un cadre théorique et méthodologique identifiant mieux la prise en compte de facteurs humains. Cette méthode, dite en U, est issue de travaux menés depuis plusieurs années au LAMIH. Ce

cadre a permis de mieux positionner les méthodes, modèles, concepts et techniques expliqués dans les parties suivantes.

La seconde partie a été consacrée à l'analyse et la modélisation des tâches humaines et de l'utilisateur, étapes insuffisamment prises en compte dans la plupart des modèles et méthodes de développement issus du génie logiciel. Elles font l'objet de recherches très actives, et débouchent progressivement sur des démarches et techniques opérationnelles.

La troisième section s'est penchée sur l'architecture de l'interface homme-machine. Deux tendances se dégagent pour les architectures recensées dans la littérature. Certaines architectures décomposent l'interface en différents niveaux logiciels visant une séparation nette entre l'interface et l'application. Même si ces architectures offrent des avantages du point de vue génie logiciel, elles n'apportent pas véritablement de solutions systématiques concernant la prise en compte des facteurs humains. D'autres interfaces, qualifiées à tort ou à raison "d'intelligentes", proposent quant à elles une prise en compte explicite d'un ensemble de facteurs liés aux tâches et à l'utilisateur, et ceci directement dans l'interface, mais éventuellement au détriment des aspects techniques. Il s'agira dans les années à venir de trouver un compromis entre ces modèles d'interfaces homme-machine. *Une de nos contributions à l'ingénierie des interfaces concerne les interfaces dites "intelligentes".*

Dans la mesure où de nombreux auteurs préconisent une démarche de conception itérative, il a semblé utile de traiter dans la dernière partie la spécification, la réalisation et l'évaluation des interfaces. De nombreux outils et méthodes ont été identifiés. Leur diversité rend encore ardue leur prise en compte systématique actuellement. Toutefois, il faut s'attendre à une unification progressive de certaines de ces méthodes et à leur intégration à terme dans les démarches du génie logiciel. *Nous avons positionné dans cette section d'autres contributions à l'ingénierie des interfaces.*

Le chapitre suivant est consacré à une synthèse de nos contributions au domaine de recherche.



## Bibliographie

- ABED M. (1990).** *Contribution à la modélisation de la tâche par outils de spécification exploitant les mouvements oculaires : application à la conception et à l'évaluation des interfaces homme-machine.* Thèse de doctorat, université de Valenciennes, Septembre.
- ABED M., ANGUE J.C. (1990).** *Using the measure of eye movements to modelize an operator's activity.* Ninth European Annual Conference on "Human decision making and manual control", Varese, Italy, September 10-12.
- ACIER B. (1990).** *Contribution à la conception et à l'évaluation d'interfaces Homme-Machine par l'analyse du comportement visuel d'un superviseur humain.* Thèse de doctorat, université de Valenciennes, Janvier.
- ACM Special Interest Group on Computer-Human Interaction (1992).** *Curriculum Development Group ACM SIGUI Curricula for Human-Computer Interaction.* ACM, New-York.
- AMALBERTI R., DEBLON F. (1992).** *Cognitive modeling of fighter aircraft process control: a step towards an intelligent onboard assistance system.* International Journal of Man-Machine Studies, 36(5).
- AMALBERTI R., De MONTMOLLIN M., THEUREAU J. (Eds.) (1991).** *Modèles en analyse du travail.* Liège : Mardaga.
- ANDERSON J.R. (1983).** *The architecture of cognition.* Cambridge : Harvard University Press.
- AUSSENAC N. (1989).** *Conception d'une méthodologie et d'un outil d'acquisition de connaissances expertes.* Thèse de doctorat, université Paul Sabatier de Toulouse, Octobre.
- AVOURIS N.M., LIEDEKERKE M.H. (1993).** *User interfaces design for cooperating agents in industrial process supervision and control applications.* International Journal of Man-Machine Studies, 38, 873-890.
- BADDELEY A.D., HITCH G. (1974).** *Working memory.* In G.H. Bower (Ed.), *The Psychology of Learning and Motivation*, New York : Academic Press.
- BALBO S., COUTAZ J. (1992).** *Une taxonomie pour les méthodes d'évaluation des interfaces homme-machine.* Congrès IHM'92, Paris.
- BARTHET M.F. (1988).** *Logiciels interactifs et ergonomie, modèle et méthodes de conception.* Dunod informatique Paris.
- BASS L. LITTLE R., PELLEGRINO R., REED S., SEACORD S., SHEPPARD S., SZESUR M. (1991).** *The Arch model: Seeheim revisited.* Proceedings of User Interface Developers' Workshop, Seeheim, April.
- BASTIEN C. (1991).** *Validation de critères ergonomiques pour l'évaluation d'interfaces utilisateurs.* Rapport de recherche, INRIA, n. 1427, Mai 1991.
- BEAUDOUIN-LAFON M. (1991).** *User interface management systems: present and future.* Proceedings of Eurographics Conference, Vienne, Autriche, September.
- BEKA BE NGUEMA M., WAROUX D., MALVACHE N. (1992).** *Contributing to the design of a human error tolerant man-machine interface during the control of a simple simulated industrial process.* 11th European Annual Conference on Human Decision Making and Manual Control, 17-19 novembre, Valenciennes.
- BENAISSA M. (1993).** *Une démarche de conception, réalisation et évaluation d'IHM : application au projet ferroviaire ASTREE.* Thèse de doctorat, université de Valenciennes, décembre.
- BENAISSA M., EZZEDINE H., ANGUE J.C. (1992).** *Conception et évaluation d'une interface homme-machine d'un procédé ferroviaire.* Forum SCGM Transport'92, Montréal, Canada, 1-5 juin.
- BENKIRANE M. (1991).** *Contribution à la méthodologie d'extraction de connaissances dans le domaine du diagnostic technique.* Thèse de doctorat, université de Valenciennes, Mars.
- BENYSH D.V., KOUBEK R.J., CALVEZ V. (1993).** *A comparative review of knowledge structure measurement techniques for interface design.* International Journal of Human-Computer Interaction, 5 (3), pp. 211-237.
- BERGER T. (1992).** *Contribution à l'étude de l'activité cognitive de l'opérateur, et à l'évaluation de sa charge de travail.* Thèse de doctorat, université de Valenciennes, mars.
- BERSINI H. (1989).** *La modélisation cognitive dans l'étude des systèmes homme-machines.* Thèse de doctorat, université libre de Bruxelles.
- BOEHM B.W. (1981).** *Software Engineering Economics.* Englewoods Cliffs N.J. Prentice Hall.
- BOEHM B.W., GRAY T.E., SEEWALDT T. (1984).** *Prototyping versus specifying: a multiproject experiment.* IEEE transactions on Software Engineering, 10 (3), May.
- BOOCH G. (1994).** *Conception orientée objets et applications.* 2ème édition, Addison-Wesley.

- BOY G. (1983).** *Etude de la charge de travail dans une cabine de pilotage. Mode opératoire et stratégie de récupération dans le système MESSAGE.* Rapport CERT DERA/AIRBUS Industrie, n. 11/7275, Août.
- BOY G. (1986).** *An Expert System for Fault Diagnosis in Orbital Refueling Operations.* 24th Aerospace Science Meeting AIAA'86, Reno, Nevada, January.
- BOY G. (1988a).** *Assistance à l'opérateur, une approche de l'intelligence Artificielle.* Editions Tekna, Octobre.
- BOY G. (1988b).** *Operator assistant systems.* In Cognitive engineering in complex dynamic worlds, edited by E. Hollnagel, G. Mancini, D.D. Woods, Academic Press.
- BOY G., TESSIER C. (1985).** *Cockpit analysis and assessment by the MESSAGE methodology.* 2nd IFAC Conference on "Analysis and Evaluation of man-machine systems", Varese, Italy, pp. 46-56, Septembre.
- BROWN C.M. (1988).** *Human-computer interface design guidelines.* Norwood, NJ : Ablex.
- CACCIABUE P.C., DECORTIS F., DROZDOWICZ B. (1990a).** *Modélisation cognitive de l'activité d'un opérateur contrôlant un système complexe.* Symposium sur la Psychologie du Travail et les Nouvelles Technologies, Université de Liège, 17-18 Mai.
- CACCIABUE P.C., MANCINI G., BERSINI U. (1990b).** *A model of operator behaviour for man-machine system simulation.* Automatica, 26 (6), pp. 1025-1034.
- CALVEZ J.P. (1990).** *Spécification et conception des systèmes : une méthodologie.* Masson, Paris.
- CARD S.K., MORAN T.P., NEWELL A. (1983).** *The psychology of human-computer interaction.* Hillsdale, NJ : Erlbaum.
- CARTER J.A. (1991).** *Combining task analysis with software engineering in a methodology for designing interactive systems.* In "Taking Software Design Seriously", J. Karat (Ed.), Academic Press.
- CELLIER J.M. (1990).** *L'erreur humaine dans le travail.* In J. Leplat et G. De Terssac (Eds), Les Facteurs humains de la fiabilité dans les systèmes complexes, Edition Octarés, Marseille.
- CHAIB-DRAA B. (1990).** *Contribution à la résolution distribuée de problème : une approche basée sur les états intentionnels.* Thèse de doctorat, université de Valenciennes, Juin.
- COAD P., YOURDON E. (1990).** *Object-oriented analysis.* Second edition, Yourdon Press, Prentice Hall.
- COCKERILL-SAMBRE (1984).** *Prise en compte des facteurs humains dès la conception et l'installation de la coulée continue de la S.A. Cockerill-Sambre-Chertal,* Action Communautaire Ergonomie, Recherche 107 247-11-026, Liège.
- COLLONGUES A., HUGUES J., LAROCHE B. (1986).** *MERISE, méthode de conception.* DUNOD Informatique, Paris.
- COUTAZ J. (1987).** *PAC, an implementation model for dialog design.* Proceedings of Interact'87 Conference, H.J. Bullinger, B. Shackel (Eds.), North Holland, September, pp. 431-436.
- COUTAZ J. (1988).** *De l'ergonome à l'informaticien, pour une méthode de conception et de réalisation des interfaces homme-machine.* Actes du Colloque ERGO-IA'88 "Ergonomie et Intelligence Artificielle", Biarritz, Octobre.
- COUTAZ J. (1990).** *Interfaces homme-ordinateur : conception et réalisation.* BORDAS, Paris.
- COUTAZ J. (1991).** *Interfaces homme-machine : un regard critique.* Techniques et Sciences Informatiques, 10 (1).
- COUTAZ J., CROWLEY J., LAUGIER C. (1992).** *Intelligence artificielle et interfaces homme-machine.* Actes des 4èmes journées sur l'intelligence artificielle, Marseille, pp. 169-186, Octobre.
- COUTAZ J., NIGAY L. (1991).** *Seeheim et architecture multi-agent.* Actes de IHM'91, Dourdan, décembre.
- CROCHART K., WALFARD D. (1991).** *Pratiques du maquettage-prototypage.* Génie Logiciel, n. 24, EC2, septembre.
- DANCE J.R., GRANOR T.E., HILL R.D., HUDSON S.E., MEADS J., MYERS B.A., SHULERT A. (1987).** *The run-time structure of UIMS-supported applications.* Computer Graphics, 21 (2), april.
- DANG W. (1988).** *Formal specification of interactive languages using definite clause grammars.* Réunion internationale sur l'implantation de langage de programmation et la programmation logique, Orléans, mars.
- DANIELLOU F. (1986).** *L'opérateur, la vanne, l'écran : l'ergonomie dans les salles de contrôle.* Montrouge, ANACT, collection "Outils et Méthodes".
- DERICQUEBOURG G. (1991).** *Contribution à l'élaboration d'une méthodologie de conception des interfaces opérateur incluant une commande visuelle.* Thèse de doctorat, université de Valenciennes, Octobre.
- DE KEYSER V., DECORTIS F., HOUSIAUX A., VAN DAELE A. (1987).** *Les communications homme-machine dans les systèmes complexes.* Rapport politique scientifique FAST n°8, université de Liège.
- DE KEYSER V. et al. (1992).** *The Nature of Human Expertise.* Rapport intermédiaire établi dans le cadre de la convention RFO/AI/18, université de Liège, Faculté de Psychologie et des sciences de l'éducation, 189 pages, Mars.

- DeMARCO T. (1979).** *Structured analysis and system specification*. Yourdon Press, NY, Englewood Cliffs - Prentice Hall.
- DIAPER D. (1989).** *Task analysis for human-computer interaction*. Ellis Horwood Limited, Chichester, United Kingdom.
- DIENG R. (1990).** *Méthodes et outils d'acquisition des connaissances*. ERGO-IA'90 "Ergonomie et Informatique Avancée", Biarritz.
- DRURY C.G. (1990).** *Methods for direct observation of performance*. In: Evaluation of Human works: a practical ergonomics methodology. Wilson J.R. and Corlett E.N. (Eds.), Taylor & Francis, pp. 890.
- DUVAL T. (1993).** *Construction des interfaces homme-machine : modèles et outils*. Thèse de doctorat, Ecole Centrale de Nantes, 27 janvier.
- EDMONDS E.A. (1981).** *Adaptative Man-Computer Interfaces*. In: M.J. Coombs and J.L. Alty (eds.), *Computing skills and the user interface*, London, Academic press.
- EL FAROUKI L., SCAPIN D., SEBILLOTTE S. (1991).** *Prise en compte des tâches du contrôleur pour l'ergonomie des interfaces*. Rapport final CENA/N.91223, Convention entre le CENA et l'INRIA, août.
- EL MRABET H. (1991).** *Outils de génération d'interfaces : état de l'art et classification*. Rapport INRIA, Rocquencourt, n. 126.
- FADIER E. (1990).** *Fiabilité humaine : méthodes d'analyse et domaines d'application*. In J. Leplat et G. De Terssac (Eds.), *Les Facteurs humains de la fiabilité dans les systèmes complexes*, Edition Octarés, Marseille.
- FALZON P. (1989).** *Ergonomie cognitive du dialogue*. Editions P.U.G., Grenoble.
- FAVERGE J.M. (1972).** *L'analyse du Travail*. Dans "Traité de Psychologie Appliquée", tome 3, pp. 5-60, Paris, PUF.
- FERBER J. (1990).** *Conception et programmation par objets*. Technologie de Pointe, Hermes, Paris.
- FISHER G.L., JOY K.I. (1987).** *A control panel for graphics and image processing applications*. Proceedings of the SIGCHI+GI'87, ACM, New York, pp. 287-290.
- FLANAGAN C. (1954).** *The Critical incident technique*. Psychology Bulletin, 51, p. 327-386.
- FOLEY J.D. (1987).** *Transformation on a formal specification of user computer interface*. Computer Graphics, avril, pp. 109-112.
- FOLEY J.D., VAN DAM A. (1982).** *Fundamentals of interactive computer graphics*. Addison-Wesley.
- FORSE T. (1989).** *Qualimétrie des systèmes complexes, mesure de la qualité des logiciels*. Les Editions d'Organisation.
- GALACSI (1986).** *Les systèmes d'information, analyse et conception*. Dunod Informatique, Paris.
- GERTMAN D.I., BLACKMAN H.S., BANKS W.W., PETERSEN R.J. (1982).** *CRT display evaluation: the multidimensional rating of CRT generated displays*. Report for the US Nuclear Regulatory Commission, Washington.
- GIBBONS S.C. (1992).** *Towards an expert system based menu interface evaluation tool*. Rapport de recherche, n. 1581, INRIA, janvier.
- GILMORE W.E., GERTMAN D.I., BLACKMAN H.S. (1989).** *User-computer interface in process control. A Human Factors Engineering Handbook*, Academic Press.
- GOLDBERG A. (1984).** *Smalltalk-80, the interactive programming environment*. Addison-Wesley.
- GOLDBERG A., ROBSON D. (1983).** *Smalltalk 80 : the language and its implementation*. Addison-Wesley.
- GOULD J.D. (1988).** *How to design usable system*. In "Handbook of Human-Computer Interaction", M. Helander (ed.), Elsevier Science Publishers B.V., North-Holland.
- GRANDJEAN E. (1983).** *Précis d'ergonomie*. Les Editions d'Organisation, Paris.
- GRAY M. (1975).** *Questionnaire typography and production*. Applied Ergonomics, 6, pp. 81-89.
- GROUPE CIC (1989).** *Guide d'utilisation d'AXIAL, tome 1 : synthèse, démarche et documentation, tome 2 : outils méthodologiques associés*. Les Editions d'Organisation.
- HAMMOND N., HINTON G., BARNARD P., MacLEAN A., LONG J., WHITEFIELD A. (1984).** *Evaluating the interface of a document processor: a comparison of expert judgement and user observation*. In Human-Computer Interaction, INTERACT'84, B. Schackel (Ed.), Elsevier Science Publishers B.V.: North-Holland, IFIP, 725-729.
- HAMMOUCHE H. (1993).** *De la modélisation des tâches à la spécification d'interfaces utilisateur*. Rapport de recherche INRIA, n. 1959, juillet.
- HASSL D.F. (1965).** *Advanced concepts in fault tree analysis*. System Safety Symposium, 8-9 Juin, Seattle.
- HATLEY D.J., PIRBHAI I.A. (1991).** *Stratégies de spécification des systèmes temps réel (SA-RT)*. Editions Masson.

- HAYES P.J., SZEKELY P.A. (1983).** *Graceful interaction through the cousin command interface*. International Journal of man-machine studies, 19 (3), september, pp. 285-305.
- HEFLEY, W.E. (1990).** *Architecture for adaptable Human-Machine Interface*. The second International Conference on "Human aspects of advanced manufacturing and hybrid automation", Honolulu, Hawaii, USA, August 12-16.
- HOC J.M. (1993).** *Ergonomie de la conduite des processus industriels - 1. Le fonctionnement cognitif de l'opérateur humain dans la conduite / supervision d'environnements dynamiques : approche théorique*. Rapport rédigé dans le cadre de la convention INRS/NEB/CNRS n. 5930400, PERCOTEC, Université de Valenciennes.
- HOC J.M., AMALBERTI R. (sous presse).** *Diagnosis : some theoretical questions raised by applied research*. International Journal of Psychology.
- HOLLNAGEL E. (1989).** *The design of fault tolerant systems : prevention is better than cure*. 2nd European Meeting on cognitive science approaches to process control, October 24-27, Siena, Italy.
- IGL TECHNOLOGY (1989).** *SADT, un langage pour communiquer*. Editions Eyrolles.
- ILOG (1989a).** *AIDA : manuel d'utilisation*. Société ILOG, Gentilly.
- ILOG (1989b).** *MASAI : manuel d'utilisation*. Société ILOG, Gentilly.
- JACKSON M. (1983).** *System Development*. Prentice Hall, 1983.
- JAMES M.G. (1991).** *PRODUSER: PROcess for Developing USER Interfaces*. In "Taking Software Design Seriously", J. Karat (Ed.), Academic Press.
- JAULENT P., LARRIEUX P. (1993).** *La méthodologie objet, une approche de l'ingénierie simultanée avec SYS\_P\_O*. Armand Colin, Paris.
- JURAIN T. (1991).** *De l'écrit à l'écran, étude et classification des aides logicielles au développement d'interfaces graphiques*. Génie logiciel et systèmes experts, 24, septembre.
- KARSENTY S., WEIKART C. (1991).** *Une extension de l'interface d'application dans le modèle de Seeheim*. Actes de IHM'91, Dourdan, décembre.
- KELLOG W.A. (1987).** *Conceptual consistency in the user interface: effect on user performance*. In Human-Computer Interaction, INTERACT'87, H.J. Bulinger and B. Shackel (Eds.), Elsevier Science Publishers B.V., North-Holland, 389-394.
- KELLOG W.A., BREEN T.J. (1983).** *Evaluating user and system models: applying scaling techniques to problems in Human-Computer Interaction*. In Human Factors in Computing Systems-IV, J.M. Carroll et P.P. Tanner (Eds.), ACM, North-Holland, Amsterdam, 303-308.
- KIERAS D.E., POLSON P.G. (1985).** *An approach to the formal analysis of user complexity*. International Journal of Man-Machine Studies, 22, pp. 365-394.
- KNAEUPER A., ROUSE W.B. (1985).** *A rule-based model for human problem solving behavior in dynamic environments*. IEEE SMC, 15 (6), November/December.
- KOLSKI C. (1993).** *Ingénierie des interfaces homme-machine, conception et évaluation*. Editions Hermès, Paris, 372 pages, août.
- LEPLAT J. (1985).** *Erreur humaine, fiabilité humaine dans le travail*. Armand Colin, Paris.
- LEQUESNE J. (1991).** *Les simulateurs dans le domaine de l'énergie*. Revue Générale de l'Electricité, n°5/91, Mai.
- LINDSAY P.H., NORMAN D.A. (1980).** *Traitement de l'information et comportement humain, une introduction à la psychologie*. Editions Etudes Vivantes, Montréal-Paris.
- LONGWORTH G., NICHOLLS D. (1986).** *SSADM Manuel, volume 1 : tasks and terms, volume 2 : technique and documentation*. NCC, The National Center for Information Technology, vers. 3.
- LÖWGREN J., NORDQVIST T. (1992).** *Knowledge-based evaluation as design support for graphical user interfaces*. CHI'92 Conference Proceedings, ACM, Addison-Wesley.
- LUND N.A. (1985).** *Evaluating the user interface: the candid camera approach*. In Human Factors in Computing Systems II, L. Borman and B. Curtis (Eds.), ACM, North-Holland, pp. 107-113.
- MACKINLAY J. (1986).** *Automating the design of graphical presentation of relational information*. ACM Transactions on Graphics, 5 (2), 110-141.
- MAGUIRE M., SWEENEY M. (1989).** *System monitoring : garbage generator or basis for comprehensive evaluation system ?* In People and Computers V, A. Sutcliffe et L. Macaulay (Eds.), Proceedings of the Fifth Conference of the British Computer Society Human-Computer Interaction Specialist Group, university of Nottingham, 5-8 September.

- MANDIAU R. (1993).** *Contribution à la modélisation des univers multi-agents : génération d'un plan partagé.* Thèse de doctorat, université de Valenciennes, Février.
- MANDIAU R., KOLSKI C., MILLOT P., (1991b).** *Perspective of human-machine system modelling by logical approach: contribution of technics issued from distributed problem solving.* 11th Congress of the International Ergonomics Association, Paris, July 15-20.
- MANDIAU R., KOLSKI C., MILLOT P., CHAIB-DRAA B. (1991a).** *A new approach for the cooperation between human(s) and assistance system(s): a system based on intentional states.* World Congress on Expert Systems, Orlando, Florida, Dec. 16-19.
- MARTIN J. (1989).** *Information Engineering: Introduction, book I.* Prentice Hall.
- MARTIN J. (1990).** *Information Engineering: Design and construction, book II.* Prentice Hall.
- MASSON M. (1994).** *Prévention automatique des erreurs de routine.* Thèse de Psychologie, Université de Liège, juillet.
- MASSON M., DE KEYSER V. (1992).** *Human errors learned from field study for the specification of an intelligent error prevention system.* In S. Humar (Ed.), *Advances in Industrial Ergonomics and Safety IV*, pp. 1085-1092, Taylor & Francis.
- McGREW J. (1991).** *Tools for task analysis: graphs and matrices.* In "Taking Software Design Seriously", J. Karat (Ed.), Academic Press.
- MEINADIER J.P. (1991).** *L'interface utilisateur, pour une informatique plus conviviale.* Dunod, Paris.
- MILLER G.A. (1975).** *The psychology of communication.* Basic Books, New York, 2nd Ed.
- MILLOT P. (1988).** *Supervision des procédés automatisés et ergonomie.* Editions Hermes, Paris, Décembre.
- MILLOT P. (1990).** *Coopération homme-machine dans les procédés industriels.* Dans "Les facteurs humains de la fiabilité dans les systèmes complexes", J. Leplat et G. De Terssac (éditeurs), Editions Octares/Entreprises, Marseille.
- MILLOT P., DEBERNARD S. (à paraître).** *Coopération homme-machine : problématique et méthodologie de mise en œuvre.* Le Travail Humain, numéro spécial intitulé "approches cognitives de la supervision d'environnements dynamiques", publication prévue pour 1994.
- MILLOT P., ROUSSILLON E. (1991).** *Man-Machine Cooperation in Telerobotics : Problematics and Methodologies.* Second France Israël Symposium on Robotics, Institut National des Sciences et Techniques Nucléaires, Gif sur Yvette, Avril.
- MIT (1987).** *X-Toolkit library, C Language Interface, X protocol version 11.* report.
- MOLICH R., NIELSEN J. (1990).** *Improving a human computer dialogue.* Communications of the ACM, 33, pp. 338-348.
- MONK A.F., DIX A. (1987).** *Refining early design decisions with a black box model.* In People and Computers III, D. Diaper and R. Winder (Eds.), Cambridge university Press, pp. 147-158.
- MORAN T.P. (1981).** *The Command Language Grammar: a representation of the user interface of interactive computer system.* International Journal of Man-Machine Studies, 15, pp. 3-50.
- NANARD J., NANARD M. (1993).** *Hypertexte à base de connaissances et génération dynamique d'interface.* Cinquièmes Journées sur l'Ingénierie des Interfaces Homme-Machine, Lyon, 19-20 Octobre 1993.
- NASSIET D. (1987).** *Contribution à la méthodologie de développement des systèmes experts : application au domaine du diagnostic technique.* Thèse de docteur-ingénieur, université de Valenciennes, novembre.
- NIELSEN J. (1993).** *Usability engineering.* Academic Press.
- NII H.P. (1986a).** *Blackboard systems: blackboard application systems, blackboard systems from a knowledge engineering perspective.* AI Magazine, 7 (2), pp. 82-106.
- NII H.P. (1986b).** *Blackboard systems: the blackboard model of problem-solving and the evolution of blackboard architectures.* AI Magazine, 7 (3), pp. 39-53.
- NORMAN D.A. (1986).** *Cognitive engineering.* In D.A. Norman & S.W Draper (Eds), *User centred system design: new perspectives on human computer interaction.* Hillsdale, NJ : Erlbaum.
- NORMAND V. (1992).** *Le modèle SIROCO : de la spécification conceptuelle des interfaces utilisateurs à leur réalisation.* Thèse de doctorat de l'université Joseph Fourier, Grenoble I.
- PAAP K.R., ROSKE-HOFSTRAND R.J. (1988).** *Design of menus.* In "Handbook of Human-Computer Interaction", M. Helander (ed.), Elsevier Science Publishers B.V., North-Holland.
- PASCOT D., BERNADAS C. (1993).** *L'essence des méthodes : étude comparative de six méthodes de conception de systèmes d'information informatisés.* INFORSID'93 "systèmes d'information, systèmes à base de connaissances", Lille, 11-14 mai.
- PERLMAN G. (1987).** *An axiomatic model of information presentation.* Proceedings of the 1987 Human Factors Society Meeting, New York, Human Factors Society, 1229-1223.

- PERLMAN G. (1988).** *Software tools for user interface development*. In "Handbook of Human-Computer Interaction", M. Helander (ed.), Elsevier Science Publishers B.V., North-Holland.
- PPAFF G.E. (1985).** *User interface management system*. Springer-Verlag.
- PHILIPPE B. (1993).** *Système d'aide à la conception des protocoles expérimentaux en communication homme-machine, fondée sur une méthodologie orientée par les objets*. Thèse de doctorat, université de Valenciennes, février.
- PINSKY L., THEUREAU J. (1985).** *Signification et action dans la conduite des systèmes automatisés de production séquentielle*. Collection d'Ergonomie et de Neurophysiologie du travail, CNAM, n. 83, Paris.
- POULAIN T. (1994).** *Contribution du génie logiciel pour la conception et l'évaluation d'applications de supervision* Thèse de doctorat, université de Valenciennes, 25 février.
- POULAIN T., KOLSKI C. (1992).** *Monde réel et monde virtuel, la problématique du contrôle de procédé par un opérateur humain*. Conférence Informatique 92 "L'interface des mondes réels et virtuels", Montpellier, 23-27 mars.
- PRADENC H. (1992).** *Superviseurs : des fenêtres ouvertes sur le process*. Revue AXES Robotique-Automatique, 61, Janvier.
- RASMUSSEN J. (1980).** *The human as a system component*. In H.T. Smith and T.R.G. Green Editors, Human Interaction with Computer, London Academic Press.
- RASMUSSEN J. (1983).** *Skills, rules and knowledge; signals, signs and symbols, and other distinctions in human performance models*. IEEE Transactions on Systems, Man and Cybernetics, volume 13, n°3, pp. 257-266.
- RASMUSSEN J. (1986).** *Information processing and human-machine interaction, an approach to cognitive engineering*. Elsevier Science Publishing.
- RAVDEN S.J., JOHNSON G.I. (1989).** *Evaluating usability of human-computer interfaces: a practical method*. Ellis Horwood, Chichester.
- REASON J. (1987).** *A framework for classifying errors*. In "New Technology and Human Error", J. Rasmussen and J. Leplat (Eds.), Wiley and Sons, Chichester.
- REASON J. (1990).** *Human error*. Cambridge University Press.
- RECHT J.L. (1966).** *Failure mode and effect*. National Safety Council, USA.
- REISNER P. (1981).** *Formal grammar and human factors design of an interactive graphics system*. IEEE Transactions on Software Engineering, 7 (2), March.
- REISNER P. (1984).** *Formal Grammar as a tool for analyzing ease of use, some fundamental concepts*. In "Human Factors in Computer Systems", J.C. Thomas, M.L. Schneider (Eds.), Ablex, New Jersey.
- Revue MESURES (1990).** *Contrôle de process, symboles graphiques et codes d'identification*, Fiche 30. Mesures, pp. 79-84, 25 Juin.
- RICHARD J.F., BONNET C., GUIGLIONE R. (Eds.) (1990).** *Traité de Psychologie Cognitive, le traitement de l'information symbolique*. Paris : Dunod.
- RIERA B. (1993).** *Methodologie d'évaluation des interfaces Homme/Vehicule automobile*. Thèse de Doctorat, université de Valenciennes, janvier.
- ROLLAND C., FOUCAUT O., BENCI G. (1988).** *Conception des systèmes d'information, la méthode REMORA*. Editions Eyrolles, Paris.
- ROUSE W.B. (1983).** *Models of human problem solving: detection, diagnosis and compensation for system failures*. Automatica, 19 (6), november.
- ROUSE W.B. (1988).** *Adaptive aiding for Human-Computer control*. Human Factors, 30 (4), pp. 431-443.
- ROUSE W.B., MORRIS N.M. (1985).** *Conceptual design of an human error tolerant interface for complex engineering systems*. 2nd IFAC Conference on "Analysis and Evaluation of man-machine systems", Varese, Italy, pp. 46-56, september.
- ROUSE W.B., ROUSE S.H. (1983).** *Analysis and classification of human error*. IEEE trans. syst. man and Cybern., SMC-13, pp. 539-549.
- RUMBAUGH J. BLAHA M., PREMERLANI W., EDDY F., LORENSEN W. (1991).** *Object-oriented modeling and design*. Prentice-Hall.
- SACERDOTI E.D. (1977).** *A structure for plans and behavior*. Elsevier, computer science library.
- SCAPIN D. (1986).** *Guide ergonomique de conception des interfaces homme-machine*. Rapport technique INRIA, n°77, Le Chesnay.

- SCAPIN D. (1990).** *Organizing human factors knowledge for the evaluation and design of interfaces*. International Journal of Human-Computer Interaction, 2 (3), pp. 203-229.
- SCAPIN D., PIERRET-GOLBREICH C. (1989).** *MAD : une Méthode Analytique de Description des Tâches*. Actes du colloque sur l'ingénierie des interfaces *Homme-Machine*. Sophia-antipolis, 24-26 Mai.
- SCHNEIDERMAN B. (1987).** *Designing the user interface: strategies for effective human-computer interaction*. Reading, MA: Addison-Wesley.
- SCHULER A.J., ROGERS G.T., HAMILTON J.A. (1985).** *ADM - A dialog manager*. Proceedings of SIGCHI'85 Conference, ACM, New York, April, pp. 177-183.
- SEBILLOTTE S. (1987).** *La planification hiérarchique comme méthode d'analyse de la tâche: Analyse de tâches de bureau*. Rapport technique INRIA, n° 599, Le Chesnay.
- SENACH B. (1990a).** *Evaluation ergonomique des interfaces Homme-machine: une revue de la littérature*. Rapport de recherche, INRIA, Sophia Antipolis, n° 1180, Mars, 70p.
- SENACH B. (1990b).** *Evaluation de l'ergonomie des interfaces homme-machine : du prototypage aux systèmes experts*. Actes du Congrès ERGO-IA '90 : Ergonomie et Informatique avancée, Biarritz, septembre.
- SHERIDAN T.B. (1988).** *Task allocation and supervisory control*. In "Handbook of Human-Computer Interaction", M. Helander (ed.), Elsevier Science Publishers B.V., North-Holland.
- SHERRIL-LUBINSKY Corporation (1990).** *SL-GMS Function Reference, Object-oriented graphical modeling system (version 4.0)*. Sherrill-Lubinsky Corporation, USA, 26 octobre.
- SIMON P. (1993).** *Contribution de l'analyse des mouvements oculaires à l'évaluation de la charge de travail mental*. Thèse de doctorat, université de Valenciennes, décembre.
- SINCLAIR M.A. (1990).** *Subjective assessment*. In: Evaluation of Human works: a practical ergonomics methodology. Wilson J.R. and Corlett E.N. (Eds.), Taylor & Francis, pp. 890.
- SMITH S.L., MOSIER J.N. (1986).** *Guidelines for designing user interface software*. Report EDS-TR-86-278, The MITRE Corporation, Bedford, MA.
- SOENEN R., LECAT J., TAHON C. (1988).** *Modelization with petri net of the dialogue*. Rapport intermédiaire, projet VITAMIN, LGIL, université de Valenciennes, juillet.
- SOMMERVILLE I. (1994).** *Le génie logiciel*. 4ème édition, Addison-Wesley.
- SPERANDIO J.C. (1988).** *L'ergonomie du travail mental*. Editions Masson, Paris.
- SPERANDIO J.C. (1991).** *Les méthodes d'analyse du travail en psychologie ergonomique*. Dans La recherche en psychologie (domaines et méthodes), J.P. Rossi (Ed.), Bordas, Paris.
- STAMMERS R. B., CAREY M. S., ASTLEY J. A. (1990).** *Task analysis*. In "Evaluation of human work. A Practical Ergonomics Methodology", Edited by Wilson J. R. and Corlett E. N., Taylor & Francis.
- STREVELER D.J., WASSERMAN A.I. (1984).** *Quantitative measures of the spatial properties of screen design*. In Human Computer Interaction : INTERACT '84, B. Shackel (Eds.), Amsterdam, North-Holland, pp. 81-89.
- STROUSTRUP B. (1988).** *What is Object-Oriented Programming*. IEEE Software, 5 (3), may.
- SWEENEY M., MAGUIRE M., SHACKEL B. (1993).** *Evaluating user-computer interaction : a framework*. International of Man-Machine Studies, 38, 689-711.
- TABORIN V. (1989).** *Coopération entre opérateur et système d'aide à la décision pour la conduite de procédés continus : application à l'interface opérateur système expert du projet ALLIANCE*. Thèse de doctorat, université de Valenciennes, Mars.
- TARBY J.C. (1993).** *Gestion automatique du dialogue homme-machine à partir des spécifications conceptuelles*. Thèse de Doctorat, Toulouse, septembre.
- TARDIEU H., ROCHFELD O., COLLETI R. (1991).** *La méthode Merise, principes et outils, 2ème édition*. Editions d'Organisation (tome 1), Paris.
- TARDIEU H., ROCHFELD O., COLLETI R., PANET G., VAHEE G. (1985).** *La méthode Merise, démarche et pratiques*. Editions d'Organisation (tome 2), Paris.
- TENDJAOUI M. (1992).** *Contribution à la conception d'interface "intelligente" pour le contrôle de procédés industriels : application au Module Decisionnel d'Imagerie*. Thèse de doctorat, université de Valenciennes, 20 novembre.
- THEUREAU J. (1992).** *Le cours d'action : analyse sémio-logique*. Berne, Peter Lang.
- TULLIS T.S. (1984).** *A Computer based tool for evaluating alphanumeric displays*. In Human-Computer Interaction: INTERACT'84, B. Schackel (Ed.), Amsterdam, North-Holland, 719-723.

- TULLIS T.S. (1988).** *A system for evaluating screen formats : research and application.* In "Advances in Human-Computer Interaction", H.R. Harston et D. Hix (Eds.), volume 2, Ablex Norwood, N.J., pp. 214-286.
- VANNESTE C., LAMBERT I. (1991).** *Méthodologie de prototypage pour la conception de logiciel centrée sur l'utilisateur.* Compte rendu de fin d'étude d'une recherche financée par le ministère de la Recherche et de la Technologie, décision d'aide 88.D.0791, mai.
- VAN DAELE A. (1988).** *L'écran de visualisation ou la communication verbale ? Analyse comparative de leur utilisation par des opérateurs de salle de contrôle en sidérurgie.* Le Travail Humain, 1 (51).
- VAN DAELE A. (1992).** *La réduction de la complexité par les opérateurs dans le contrôle de processus continus, contribution à l'étude du contrôle par anticipation et de ses conditions de mise en oeuvre.* Thèse de doctorat en Psychologie, université de Liège, octobre.
- VI. CORPORATION (1991).** *DATAVIEWS : reference manual, version 9.0.* Amherst Research Park, Amherst, Massachusetts 01002, U.S.A., June.
- VILLEMEUR A. (1988).** *Sûreté de fonctionnement des systèmes industriels : fiabilité, facteur humain, informatisation.* Eyrolles, Paris.
- VOEGTLIN M. (1991).** *Procédé de génération d'objets pour la création d'appllications interactives adaptables.* Thèse de doctorat, université de Haute Alsace.
- WILLEM R., BILJON V. (1988).** *Extending petri nets for specifying man-machine dialogues.* International Journal of Man-Machine Studies, 28.
- WILLIGES R.C., WILLIGES B.H., ELKERTON J. (1987).** *Software Interface Design.* In G. Salvendy (ed.), Handbook of Human factors, pp. 1416-1449, New york, Wiley.
- WILSON J.R. (1990).** *A framework and a context for ergonomics methodology.* In: Evaluation of Human works: a practical ergonomics methodology. Wilson J.R. and Corlett E.N. (Eds.), Taylor & Francis.
- WILSON J.R., CORLETT E.N. (Eds.) (1990).** *Evaluation of human works: a practical ergonomics methodology.* Taylor & Francis.
- WOODS D.D., ROTH E.M. (1987).** *Models of cognitive behaviour in nuclear power plant personnel.* NUREG/CR-4532.
- YOURDON E., CONSTANTINE A. (1979).** *Structured design.* Prentice Hall.



## Chapitre 2

### *Contribution aux méthodes et modèles de conception et d'évaluation des interfaces homme-machine*

*Activités de recherche et d'encadrement doctoral*

Dans le premier chapitre, un état de l'art a été dressé des méthodes et modèles actuels de conception et d'évaluation d'interfaces homme-machine, tout en prêtant attention à ceux concernant les systèmes industriels complexes. Dans cet état de l'art, plusieurs de nos contributions ont été positionnées. Il est utile dans un premier temps de les résumer :

- Mes premiers travaux menés en ingénierie des interfaces ont débouché sur ma thèse de doctorat en janvier 1989. Ils ont été à la base de ma recherche dans son esprit "homme-machine". Ils consistent en l'étude et la mise en œuvre d'une approche d'évaluation "automatique" d'interface, appliquée à une maquette appelée SYNOP intégrant des bases de connaissances ergonomiques. Ce travail a fait l'objet de deux études industrielles, la première dans le cadre du projet ALLIANCE et la seconde en collaboration avec RENAULT.
- D'autres travaux se sont penchés sur la génération "automatique" de spécifications d'interface. Ils ont mené à un système appelé ERGO-CONCEPTOR se composant de trois modules. Le premier permet la description de l'application en termes de composants et de sous-systèmes, et selon différents niveaux d'abstraction. Le second module exploite les données décrites précédemment pour générer "automatiquement" des spécifications d'interface. A l'aide du troisième module, il est possible au développeur d'utiliser ces spécifications pour générer interactivement l'imagerie.

- Nous nous sommes impliqués dans le prototypage d'imagerie industrielle. Ces travaux concernent l'étude et la mise en œuvre d'un outil informatique de prototypage intégré dans un atelier de génie logiciel facilitant la spécification, la réalisation et l'évaluation d'imageries. Cet outil permet de décrire interactivement l'application à l'aide de réseaux de composants, puis de spécifier les images en exploitant directement la description de l'application. Cette recherche a été réalisée en collaboration industrielle avec la CSEE et 3IP.
- Un autre volet de ce domaine de recherche a concerné les interfaces dites "intelligentes". Des travaux ont été axés sur l'étude et la mise en œuvre d'une approche d'interface "intelligente" appelée Module Décisionnel d'Imagerie (M.D.I.). Celle-ci vise à expliciter les concepts "QUOI" présenter" (correspondant au choix des informations à présenter aux utilisateurs), "QUAND" et "COMMENT" les présenter. Certains des concepts ont été validés au départ de cette recherche lors du projet ALLIANCE et d'une collaboration industrielle avec S2O.
- Enfin, en complément aux travaux précédents, des collaborations industrielles avec GAZ de FRANCE, B+ Development et SOLVAY ont conduit à la mise à disposition de développeurs d'interfaces de connaissances livresques et didactiques.

Ce second chapitre détaille maintenant chacune de ces contributions, mais selon des points de vue différents : (i) un point de vue *logiciel*, descriptif des approches, des principes, des concepts et des techniques mis en œuvre, (ii) un point de vue *méthode*, puisque chacun des systèmes réalisés est associé à une méthode de conception et/ou d'évaluation particulière.

## 2.1. DESCRIPTION DES SYSTEMES A LA BASE DE NOTRE RECHERCHE

Certains des principes et méthodes de conception et d'évaluation d'interfaces homme-machine que nous avons étudiés ces dernières années se sont retrouvés dans quatre systèmes principaux. Il est important de les présenter.

### 2.1.1. Système d'évaluation automatique d'interface homme-machine

Nos premiers travaux sur le thème de l'ingénierie des interfaces homme-machine ont commencé en septembre 1985, et ont porté sur le développement d'un système expert appelé SYNOP. Bien que ce système ait été développé dans le cadre du doctorat (KOLSKI, 1989), il a eu énormément d'importance pour tous les travaux qui ont suivi, de par l'état d'esprit homme-machine qui s'en dégageait, les méthodes de travail adoptées et l'originalité de cette démarche d'ingénierie centrée sur l'utilisateur.

Développé au L.A.I.H. et actuellement sous la forme d'une maquette, SYNOP est écrit en LISP et est interfacé avec un progiciel de création et d'animation de vues graphiques. SYNOP intègre dans ses bases, des connaissances ergonomiques "statiques" de présentation d'information. Elles sont formalisées en règles de production exploitables par un moteur d'inférence conçu par GRZESIAK (1987).

Cette section se compose de trois parties. La structure de la chaîne logicielle intégrant le système SYNOP est d'abord présenté, ainsi que son principe d'évaluation. Puis, une validation de SYNOP est discutée.

#### 2.1.1.1. Structure de la chaîne logicielle intégrant SYNOP

SYNOP dispose de deux interfaces homme-machine gérées par un superviseur. Une interface utilisateur permettant au développeur des vues de lancer et de contrôler le module d'évaluation de vues, et

une interface expert destinée à l'introduction et à la mise à jour des règles ergonomiques dans la base de connaissance, par l'ingénieur cognitif. La base de connaissance est structurée d'une part en plusieurs "sous-bases" de connaissance regroupant chacune des règles relatives à un même thème ergonomique, et d'autre part en plusieurs "sous-bases" de métaconnaissance contenant des critères de sélection des "sous-bases" de connaissance (KOLSKI, GRZESIAK et MILLOT, 1988).

SYNOP comprend un module d'évaluation "statique" de vues graphiques dont l'objectif est d'évaluer et de corriger des ébauches de vues selon des règles rangées dans sa base de connaissance, figure 2.1. Les ébauches de vues sont réalisées au moyen du poste de création graphique et stockées sous forme de fichiers. Une interface entre l'éditeur graphique et le module d'évaluation interprète les fichiers en un réseau sémantique d'objets descriptifs de la vue, à des fins d'évaluation ergonomique et de correction par le module d'évaluation. Les vues ainsi corrigées sont ensuite recodées par l'interface sous forme de fichiers, elles peuvent être alors affichées sur l'écran au moyen de l'éditeur. Au début du projet, le choix s'est porté sur le progiciel IMAGIN (SFERCA, 1987).

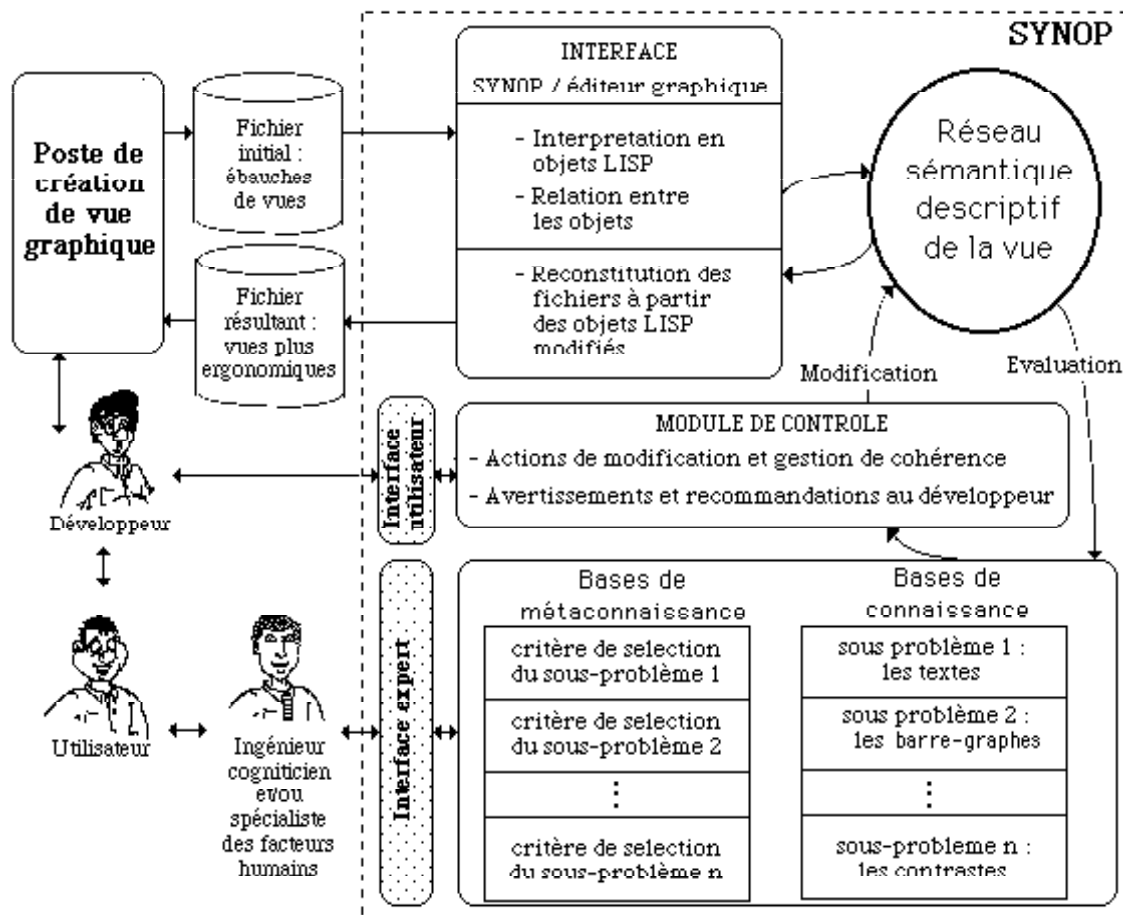


Figure 2.1 : Schéma de principe de l'évaluation statique de vues graphiques

### 2.1.1.2. Principe d'évaluation d'une vue graphique

Le principe d'évaluation et d'amélioration ergonomiques d'une vue graphique par SYNOP peut être décrit schématiquement par trois phases. La première consiste à interpréter les fichiers, créés automatiquement par le progiciel IMAGIN, relatifs à la vue à traiter afin de créer un réseau sémantique d'objets structurés. Leur structure est inspirée de la notion de frame (MINSKY, 1975). Elle comprend le nom de l'objet, par exemple "texte7", un ensemble d'attributs le caractérisant, tel que sa "couleur", ainsi que des relations avec d'autres objets, par exemple "à-l'intérieur-de".

La seconde phase est pilotée par le module de contrôle, moteur d'ordre 1, et concerne l'évaluation des objets du réseau. Elle commence par une analyse globale en vue de la sélection de "sous-bases" de connaissance. Chacune de ces "sous-bases" est associée à un concept ou thème, et contient ainsi un ensemble de règles relatives chacune à l'étude d'un sous-problème ergonomique. Par exemple, une "sous-base", concernant l'ergonomie de présentation des barre-graphes, regroupe l'ensemble des règles ergonomiques relatives à cette classe d'objets graphiques. De même, une autre "sous-base" peut concerner la lisibilité des caractères, les contrastes colorés, etc (BINOT, KOLSKI et al., 1987 ; KOLSKI, BINOT et al., 1988).

La sélection des sous-bases est effectuée par l'activation de critères de sélection sous la forme de métarègles appartenant aux sous-bases de métaconnaissance, figure 2.1 (PITRAT, 1990). Cette méthode permet de restreindre l'espace de travail, en limitant l'étape de sélection des règles activables à des ensembles de règles effectivement activables, et de réduire le temps de traitement en évitant de tester des règles inutiles. Dans l'exemple de métarègle donné ci-dessous, x1 est une variable prenant pour valeur le nom de la vue traitée. S'il existe au moins un barre-graphe dans la vue, la sélection de bases traitant le problème de la présentation des barre-graphes est proposée au développeur.

*SI (il-existe-au-moins-un-barre-graphe-dans-la-vue x1)*  
*ALORS (sélectionner-la-base-sur-les-barre-graphes x1)*

Puis, la vue est évaluée par inférence des règles d'action contenues dans chaque "sous-base" ainsi sélectionnée. Un exemple simple de règle d'action conduisant à une modification physique d'objets graphiques est présentée ci-dessous. Elle permet de séparer deux textes parallèles x1 et x2 d'une distance égale à 100% de la hauteur de leurs caractères, en cas de distance insuffisante.

*SI (2-textes-non-séparés-distance-100%-hauteur x1 x2)*  
*ALORS (séparer-d'une-distance-de-100%-hauteur x1 x2)*

L'activation d'une règle d'action peut entraîner une modification automatique des caractéristiques d'objets statiques et animés de la vue nécessitant, éventuellement, la gestion de la cohérence de l'environnement des objets modifiés. Cette gestion de cohérence s'effectue par l'activation de règles d'action particulières appelées règles de cohérence.

Dans certains cas, une modification physique d'objets graphiques peut s'avérer difficile ou impossible, suite à l'activation d'une règle d'action. C'est par exemple le cas où il n'existe que des règles ergonomiques imprécises s'appliquant à l'objet. Dans ce cas, le système se contente de fournir au développeur des recommandations ergonomiques générales concernant la vue traitée. De telles règles peuvent s'avérer imprécises en ce sens qu'elles utilisent des critères d'évaluation subjectifs. Par exemple, la règle ci-dessous recommande au développeur de ne pas surcharger la vue x1 par des zones colorées lorsque la surface totale de ces zones est supérieure à un seuil subjectif pré-défini dans la règle.

*SI (aire-zones-colorées-grande x1)*  
*ALORS (conseiller-au-développeur-réduire-aire-zones-colorées x1)*

Après cette seconde phase d'évaluation et d'amélioration de la vue, la troisième phase permet de reconstituer les fichiers graphiques correspondant à la vue améliorée, de façon à pouvoir les afficher sur l'écran. Ils sont immédiatement utilisables par le développeur. De plus, le système fournit des recommandations lui permettant de retoucher son synoptique d'après les conseils découlant de l'inférence des règles de recommandations. Après cette phase, le traitement de SYNOP est terminé. Le développeur peut alors quitter l'environnement logiciel afin d'améliorer son synoptique à l'aide de l'éditeur graphique, selon les recommandations fournies, ou relancer un autre traitement.

### 2.1.1.3. Validation de SYNOP

SYNOP a été utilisé lors de l'évaluation "statique" de l'interface homme-machine d'un système d'aide à la décision, gérant les alarmes, et utilisant des techniques d'intelligence artificielle. Cette interface a été développée lors du projet ALLIANCE (TABORIN et al., 1986, 1988 et 1990).

Ce projet a eu pour support la station d'Essais Technologique de Cadarache, qui est un simulateur physique de centrale nucléaire, et avait pour objectif la réalisation d'un système expert temps réel de traitement des alarmes. Ce projet réunissait plusieurs partenaires industriels et laboratoires. La contribution du L.A.I.H. au projet concernait la définition et la réalisation de l'interface homme-système d'aide à la décision. Cette interface devait être capable d'apporter une aide à l'utilisateur durant les deux classes de fonctionnement possibles, normale et anormale, et de lui fournir des informations lui permettant de prévoir l'état de la centrale. Des vues spécifiques ont été réalisées :

- des vues synthétiques permettant aux utilisateurs d'effectuer une tâche de surveillance sur un ensemble restreint de variables, les plus significatives de l'état de fonctionnement normal, et par extension caractéristiques de l'apparition d'un défaut.
- des vues présentant sur un synoptique simplifié de l'installation les variables impliquées dans le défaut. Ces variables étaient reliées entre elles par des arcs orientés symbolisant la propagation du défaut depuis son origine jusqu'aux variables concernées par des conseils de prévention fournis par le système expert.
- des vues de tendance affichant trois variables au maximum, parmi les plus significatives de la prédiction faite par le système expert. Chaque variable est représentée par trois courbes : l'historique de la variable, la courbe de prédiction qui représente son évolution prédite, et une courbe de suivi qui correspond à l'évolution réelle de la variable entre deux prédictions.

La présentation d'information présente sur chacune des vues développées a été évaluée par SYNOP. La connaissance implantée dans les bases de SYNOP se partageait en (KOLSKI, VANDAELE et al., 1988):

- des connaissances générales utilisables pour l'évaluation de la plupart des applications graphiques. Celles-ci constituaient la base de l'ergonomie de présentation d'information et comprenaient par exemple des règles sur la taille des caractères, sur l'emplacement des zones stratégiques sur l'écran ou le choix des couleurs pour favoriser des "bons" contrastes.
- des connaissances spécifiques à des applications industrielles particulières, où le sens et l'utilisation d'une fonction animée pouvaient différer d'une application à une autre : par exemple, pour l'application ALLIANCE, l'utilisation de la couleur jaune pour chaque variable correspondant à l'état prévu du système.

De plus, chaque connaissance mise en œuvre pouvait être associée à des connaissances "de bon sens" assurant la cohérence de l'image en cas de modification graphique d'un objet. Différentes règles étaient relatives : (i) au codage de l'information, (ii) à l'organisation et à la structuration de l'information, (iii) à la simplification et à l'allégement des vues, (iv) à la perception et l'interprétation de l'information. Une centaine de règles d'action et de recommandation ont été inférées.

Lors du traitement, de nombreuses modifications ont été effectuées automatiquement, déchargeant ainsi le développeur de tâches de correction souvent fastidieuses. Quant aux corrections restant à effectuer, et correspondant aux recommandations fournies par SYNOP, elles ont pu être réalisées à l'aide de l'éditeur graphique. Comme nous l'avons déjà souligné, les règles mises en œuvre étaient soit générales, soit spécifiques. Les règles générales pouvaient être utilisées lors de l'évaluation d'autres interfaces. Quant aux

règles spécifiques, elles étaient susceptibles d'être réutilisées ultérieurement selon l'évolution du projet pour évaluer des vues nouvelles.

#### **2.1.1.4. Conclusion sur l'approche**

Cette section a présenté notre première contribution à l'ingénierie des interfaces homme-machine, sous la forme d'un système expert original, appelé SYNOP, d'aide à l'évaluation "statique" de synoptiques industriels. Ce système a contribué à la conception de l'interface développée dans le cadre du projet ALLIANCE. Les nombreuses règles inférées lors du traitement ont débouché sur l'amélioration ergonomique des vues constituant l'interface. Les premiers résultats obtenus ont été encourageants. Bien entendu, de nombreuses règles ergonomiques devraient encore être introduites dans les bases, afin que le système puisse être appliqué sur la plupart des types d'interfaces graphiques existants. A cet égard, des travaux en relation avec RENAULT ont été menés ensuite en vue de l'évaluation ergonomique d'images embarquées dans les automobiles (KOLSKI, MOUSSA et MILLOT, 1988 ; 1989).

Notre objectif n'a pas été d'en faire un produit, compte tenu de l'incomplétude actuelle de la base de connaissances. Mais cette recherche a permis de prouver la faisabilité du concept, même si cette approche d'ergonomie de correction s'avère lourde à gérer, surtout du point de vue de la mise à jour des connaissances dans la base du système (KOLSKI et MILLOT, 1989). En tirant les leçons de cette expérience enrichissante, la conception, la réalisation et l'évaluation de SYNOP, ainsi que l'acquisition de connaissances ergonomiques ont servi de base de réflexion pour les projets décrits dans les sections suivantes.

En conclusion, un outil comme SYNOP ne remplacera pas un spécialiste des facteurs humains, ce n'est d'ailleurs pas l'objectif d'un tel système. Par contre, moyennant des aménagements, il peut lui être d'une grande utilité en le déchargeant d'un ensemble de tâches d'évaluation et de modification fastidieuses, en lui facilitant la structuration ou la mise à jour des connaissances nécessaires à l'évaluation d'une interface, ou encore en contribuant à la formation de nouveaux évaluateurs (KOLSKI et MILLOT, 1991).

### **2.1.2. Système de génération automatique de spécifications d'interface**

Nous avons contribué à l'étude et au développement d'une approche d'ingénierie pour *la conception* d'interfaces. Elle se situe dans la lignée de la précédente. Elle se présente sous la forme d'un système appelé ERGO-CONCEPTOR, actuellement à l'état de maquette, écrite en LISP, AIDA et MASAI, développée dans le cadre de la thèse de Faouzi MOUSSA (1992). Ce système est original dans la mesure où il permet de baser la conception sur la génération automatique de spécifications d'imagerie. La démarche suivie à l'aide d'ERGO-CONCEPTOR pour arriver à la génération effective de l'imagerie est d'abord résumée. Puis, la structure logicielle du système est expliquée. Par la suite, une première validation technique est décrite.

#### **2.1.2.1. Présentation de la démarche de conception visée**

La démarche sur laquelle se base ERGO-CONCEPTOR a pour objectif de fournir, à partir d'une part des caractéristiques techniques du système complexe dont on veut construire les interfaces de supervision et de ses objectifs, et d'autre part des résultats issus de l'analyse de la tâche, des spécifications de l'imagerie incluant des facteurs ergonomiques, utiles à la génération interactive des vues (MOUSSA, KOLSKI et MILLOT, 1990a ; 1990b).

La génération des spécifications doit s'appuyer sur des connaissances ergonomiques de différents niveaux d'abstraction, permettant d'aboutir à des vues graphiques répondant *a priori* à un ensemble de critères ergonomiques de base. Trois étapes sont prévues dans ce but (MOUSSA et KOLSKI, 1991).

*La première étape* consiste à décrire le système complexe, avec pour optique d'arriver à un premier niveau de données nécessaires à la spécification de l'interface. Cette description suit une démarche orientée vers les besoins informationnels des utilisateurs. Elle part de l'hypothèse que la description peut s'effectuer selon trois axes complémentaires : une description en terme de hiérarchie d'abstraction "moyens-objectifs", une description structurelle et une description fonctionnelle. La description structurelle est réalisée au moyen de regroupements successifs de sous-systèmes.

Au niveau le plus bas, un sous-système est un élément indécomposable représenté par un Groupement Fonctionnel Simple (GFS), comprenant des composants reliés entre eux. Un Groupement Fonctionnel Composé (GFC) représente un regroupement de sous-systèmes formant un sous-système de plus haut niveau. La description fonctionnelle est réalisée grâce à la représentation des interactions entre les différents sous-systèmes qui le composent, par des liens d'influence regroupés sous forme d'un réseau de causalité. Le fonctionnement d'un sous-système est décrit par des variables interconnectées reflétant son état. La description est effectuée en terme d'abstraction hiérarchique moyen/objectif afin de permettre aux intervenants de passer par les étapes identifiées par RASMUSSEN (1980) lors de la résolution d'un problème. Les informations exprimeront progressivement le passage d'une connaissance particulière et précise, à une connaissance synthétique relative au fonctionnement lié à l'objectif général du système (figure 2.2).

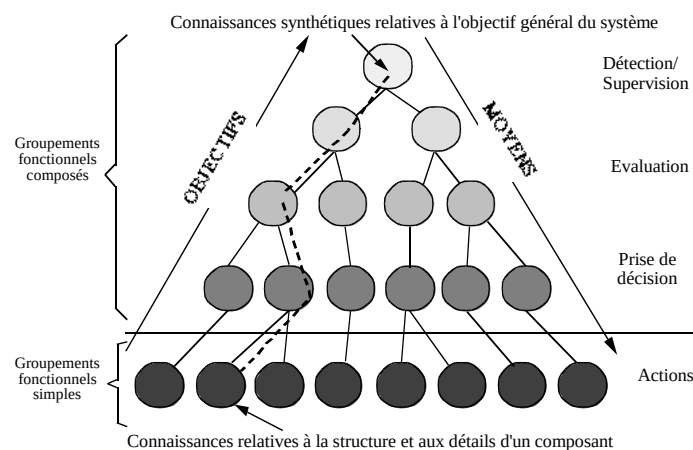


Figure 2.2 : Description en terme de hiérarchie d'abstraction moyen/objectif

*La deuxième étape* exploite les données précédentes (GFS, GFC, variables en relation), représentatives des besoins informationnels, pour spécifier automatiquement l'interface. A cet effet, ces données doivent être réparties en vues, les vues peuvent être structurées en zones et les zones doivent être composées d'entités informationnelles. A ce niveau, l'objectif est d'utiliser une base de connaissances ergonomiques pour le choix des attributs graphiques, des modes de représentation, des entités informationnelles et des structures des vues adéquates. En effet, il est possible de faire intervenir un ensemble de connaissances formalisables à l'aide de règles de production (KOLSKI, MOUSSA et MILLOT, 1990) :

- Il faut associer à chaque vue une structure adéquate et cohérente dans toute l'imagerie. Une base de connaissances peut être consacrée au choix des attributs graphiques et des modes de représentation de ces entités informationnelles. Un ensemble de règles de production peut couvrir les décisions à prendre dans ce domaine.
- Chaque mode de représentation est adapté à certaines utilisations et pas du tout à d'autres. En se basant sur les connaissances dans ce domaine, il est possible de déduire automatiquement les modes de représentation à utiliser pour une vue, à partir d'un ensemble réduit de règles de production. Cette déduction peut s'appuyer sur la description en termes de GFS, GFC, variables et liens d'influence, puisque cette description laisse explicitement supposer des besoins en termes de

supervision, d'évaluation de la situation, de commandes à effectuer, d'historiques à visualiser, etc.

- Les attributs graphiques : il s'agit de déterminer, à partir de recommandations ergonomiques, l'aspect statique de l'information à présenter à l'écran. On définira ainsi les polices de caractères, les types de lignes, les couleurs, etc.

L'ensemble des spécifications d'interface issues de cette étape doivent pouvoir ensuite être exploitables dans un but de génération de celle-ci. La génération des vues peut se faire à partir d'une base de données de spécifications qui décrit l'ensemble des vues graphiques à générer, avec tous les détails concernant leur structure, la composition des différentes zones ainsi que les attributs graphiques à réaliser. Au moment de la création, un générateur de vues graphiques à haut niveau d'abstraction doit être proposé au développeur de l'imagerie. Ce générateur peut être dit à haut niveau d'abstraction dans la mesure où la génération des spécifications doit intégrer des connaissances allant du plus haut niveau (à savoir la structuration de l'imagerie en fonction des sous-systèmes) au niveau le plus bas, c'est-à-dire les attributs graphiques composant chaque vue.

Lors de sa tâche de création graphique, le développeur doit pouvoir garder les vues graphiques générées par le système, porter des modifications sur celles-ci voire même les rejeter et se livrer à une création graphique personnalisée. De cette manière, un maximum de degrés de liberté doit lui être laissé pour éviter une éventuelle frustration due à l'utilisation d'outils de création graphique trop rigides. En d'autres termes, il devra être assisté "intelligemment" lors de la génération des vues, et non dirigé. Ce principe est proposé par le système ERGO-CONCEPTOR, décrit maintenant.

### ***2.1.2.2. La structure logicielle d'ERGO-CONCEPTOR***

ERGO-CONCEPTOR comprend trois modules mettant en oeuvre les principes décrits précédemment. Ces trois modules sont visibles sur la figure 2.3 (KOLSKI et MOUSSA, 1991).

Le premier module sert à décrire un système industriel complexe selon la démarche proposée. Il permet de gérer les variables, les GFS, les GFC et le réseau d'influence, à l'aide d'une interface dédiée, appelée sur cette figure interface "de description". Ce module est utilisable par un non-informaticien. Il peut donc être pris en charge par le développeur, un spécialiste des facteurs humains, voire même un spécifieur expert du système. Dans la première version de ce module, la description se fait essentiellement au moyen de pages-écrans alphanumériques. Elle est enregistrée dans une base de données qui servira par la suite à la génération des spécifications.

Le second module permet, selon une démarche originale, la génération automatique de spécifications d'interface en exploitant les données décrites précédemment, et à l'aide de connaissances ergonomiques stockées dans une base de connaissances. Cette base de connaissance a été préalablement alimentée à partir de l'interface dite "ergonome".

Dans ce but, toutes les données sont stockées sous forme de listes LISP qui alimenteront la base de faits d'un moteur d'inférence. Ce dernier permet de fournir des réponses sur les aspects de contenu informationnel et de présentation des vues graphiques à générer. De cette manière, on assure, tout au long de la construction des vues, une assistance permettant d'éliminer certains risques d'erreurs ou d'incohérence. La formalisation des connaissances consiste en l'énoncé de règles de présentation graphique d'informations et de structuration de vues et en leur formalisation en règles de production. Elles sont exploitées par un moteur d'inférence dont les conclusions donnent lieu à un fichier de spécification des vues graphiques à générer, respectant une syntaxe pré-définie. Cette syntaxe représente en quelque sorte une définition formelle des entités de base qui serviront lors de la construction du fichier de spécifications. Cette syntaxe présente deux avantages :



- définir à priori l'ensemble des entités manipulées, en allant du plus simple (attributs graphiques : couleur, taille, police de caractères...) au plus complexe (type de vue, structure de vue, mode de représentation, etc).
- assurer l'indépendance entre le module de description et l'éditeur graphique qui pourrait être modifié ou remplacé. En effet, le recours à une syntaxe permet d'éviter tout problème d'incohérence entre la structure informatique des données traitées par le générateur de spécification et le format "particulier" des données traitées par l'éditeur graphique choisi.

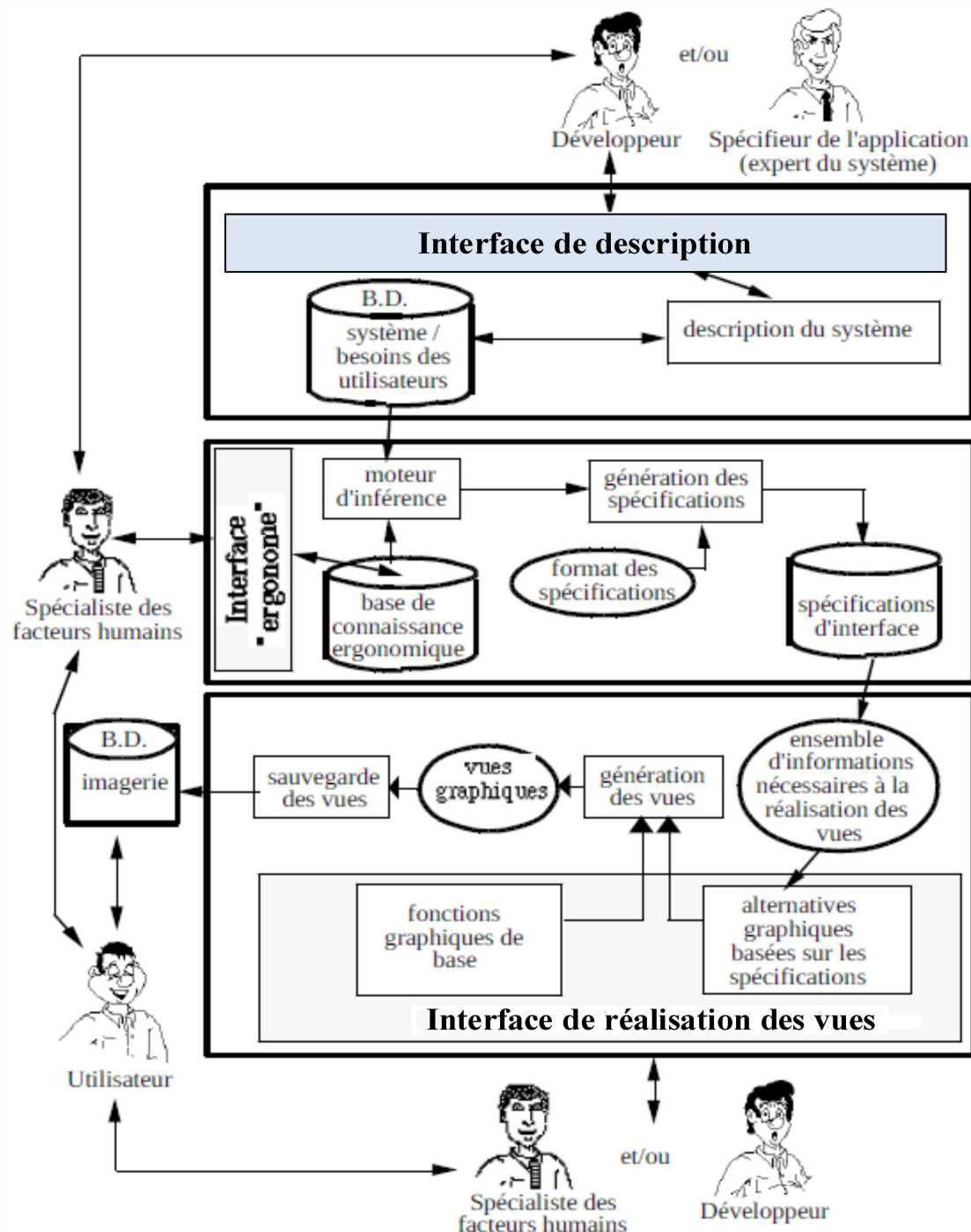


Figure 2.3 : Schéma de principe de l'utilisation d'ERGO-CONCEPTOR

Le contenu du fichier de spécifications ainsi obtenu décrit pour chaque vue : son type (supervision, commande...), le sous-système concerné, les zones que la vue englobe (définition structurelle) et le contenu de chaque zone (entités informationnelles). Quelques exemples simplifiés de règles sont visibles ci-dessous:

```

SI GFC
ALORS    vue-de-supervision ET vue-de-causalité ET vue-historique

SI vue-de-supervision ET nb-variables-inférieur-10
ALORS    vue-étoile

SI vue-à-crée
ALORS    zone-titre ET zone-alarme ET zone-travail ET zone-navigations

```

La première règle exprime que chaque GFC nécessitera la création de vues de supervision, de vues de causalité et de vues "historique". La seconde signifie que si l'on est en train de créer une vue de supervision et que cette vue pourra comporter moins de 10 variables, alors le mode graphique "étoile" est possible. La troisième spécifie que chaque vue à créer pourra être composée des zones suivantes : titre, travail, alarme et navigation.

Comme on peut le remarquer sur ces exemples, les connaissances de la base concernent la structuration des vues, le choix des attributs graphiques (couleurs, polices de caractères...) et des modes de représentation (étoile, barre-graphe...). Le type de connaissance est inféré suivant le niveau d'abstraction du processus de construction des vues. Lors de la phase de construction d'une vue, on détermine d'abord la structure adéquate pour cette vue dans le contexte de l'application traitée. Par la suite, à un niveau d'abstraction plus bas (construction de la zone de travail par exemple), on choisit de la même manière le ou les modes de représentation adéquats pour les variables à afficher. Au niveau d'abstraction le plus bas, il s'agit de préciser les attributs graphiques à utiliser.

A partir du fichier de spécifications, qui est généré par le module précédent, un générateur interactif de vues graphiques est mis à la disposition du développeur pour exploiter l'ensemble des informations décrivant la structure et les fonctionnalités des vues. Ce générateur propose au développeur, sous forme de texte accessible par des menus appelés "sous-système", "type de vue", "zones à créer", "modes de représentation" et "liste des variables", les spécifications nécessaires pour l'aider à générer ses vues. Il propose également au développeur des fonctions d'édition graphique "classiques" (primitives de dessin, fonctions d'édition de texte, annulation d'une action, effacement de l'écran, etc) lui assurant un degré de liberté important. Le développeur peut ne pas suivre toutes les propositions du générateur automatique de vues. Il a la possibilité de modifier à tout moment une partie de la vue générée automatiquement, et d'ailleurs crée lui-même le fond de plan.

Le système ERGO-CONCEPTOR a été développé à l'aide du langage LELISP et de l'environnement de développement d'applications graphiques constitué de la boîte à outils AIDA et du générateur d'interfaces MASAI, sur station graphique VAX 3100.

### **2.1.2.3. Première validation technique d'ERGO-CONCEPTOR**

Dans un but de validation, ERGO-CONCEPTOR a fait l'objet d'une première application, à partir des plans d'un système industriel de type centrale nucléaire simplifiée, accessibles principalement dans "Les Techniques de l'Ingénieur" (BEBIN, 1984). Il nous a fallu *simuler* l'étape d'analyse et de modélisation du système homme-machine (MOUSSA, KOLSKI et MILLOT, 1992). Cette simulation en laboratoire nous a conduit à un ensemble de données décrivant celui-ci.<sup>11</sup>

<sup>11</sup> Ces données n'ont pas été validées par des experts. Même si certaines propositions de décomposition du système en sous-systèmes peuvent paraître "surprenantes" pour un lecteur averti, rappelons que notre objectif est de prouver essentiellement la validité de la démarche, et qu'il est possible, à partir d'une description, d'arriver à une génération interactive assistée. Bien entendu, si la description s'avère fautive, les vues générées ne répondront pas aux besoins informationnels des utilisateurs, ce qui doit être mis en évidence lors de l'évaluation.

La démarche de description a été appliquée sur ce système. Nous sommes, bien entendu, partis sur l'hypothèse que cette description se basait sur les résultats de l'analyse préalable effectuée conjointement par les experts et les spécialistes des facteurs humains. A l'aide du premier module d'ERGO-CONCEPTOR, chacun des équipements principaux de la centrale a été décrit par un groupement fonctionnel simple. Ainsi, nous sommes partis sur l'hypothèse que dix GFS ont été recensés par les experts. Ils ont été progressivement décrits, puis rassemblés dans des GFC.

La description hiérarchique par niveau d'abstraction, illustrée par la figure 2.4, a été définie sous forme d'un arbre dont les nœuds représentent les GFC et les feuilles indiquent les GFS. Chaque niveau dans l'arbre représente un changement de niveau d'abstraction donnant lieu à des points de vue différents de la centrale. En effet, vue sous l'angle considéré, la centrale apparaît comme une arborescence où la racine "centrale nucléaire" reflète une connaissance générale globale relative à l'objectif général du système et où les feuilles (les GFS) présentent une connaissance profonde relative à la structure et aux détails des composants. La description fonctionnelle a consisté à déterminer, pour chaque GFS et GFC, les liens d'influence entre variables.

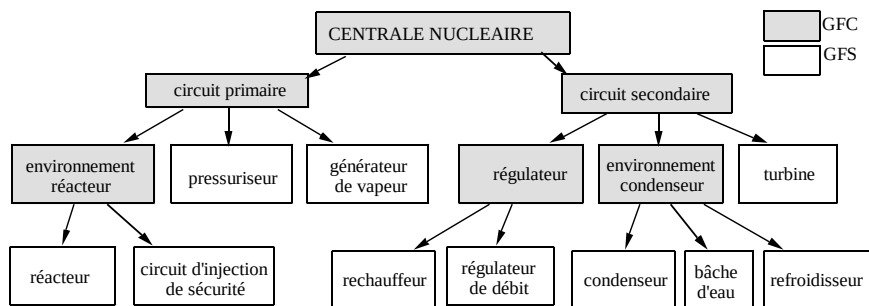


Figure 2.4 : Description hiérarchique par niveau d'abstraction

Une fois le système décrit à l'aide du premier module d'ERGO-CONCEPTOR, la génération des spécifications a été activée au moyen d'un bouton sur l'écran. Sa durée a été de quelques minutes. La génération des vues graphiques est alors devenue possible. Afin de générer une vue graphique, le développeur suivait les étapes suivantes. Il sélectionnait un sous-système parmi ceux qui lui étaient proposés en haut de l'écran, en activant le bouton "*Sous-système*" de l'éditeur d'ERGO-CONCEPTOR (figures 2.5 et 2.6), et sélectionnait dans le défileur le nom du sous-système devant faire l'objet de l'édition.

Une fois le sous-système choisi, il sélectionnait le type de vue à générer à partir du sous-système en question. Pour ce faire, il utilisait le bouton "*types de vue*" et sélectionnait parmi la liste des types proposés celui qui l'intéressait. Le sous-système et le type de vue étant connus, il s'agissait selon la même démarche de sélectionner la zone à créer, un mode de représentation et les variables à afficher. Lors de chacune des étapes citées, il pouvait utiliser des fonctions d'édition disponibles pour effectuer des modifications, telles que changer la couleur d'une zone, modifier le type de ligne d'un cadre, représenter un fond de plan graphique, etc.

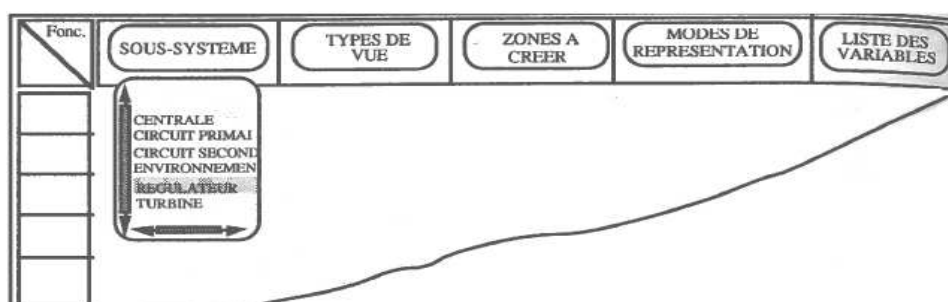


Figure 2.5 : Sélection d'un sous-système pour lequel des vues sont à créer

Pour mener à bien cette validation, nous sommes partis d'une application pour laquelle nous ne disposions pas d'expert. C'est pourquoi nous nous sommes basés sur des hypothèses concernant la décomposition de celui-ci en sous-systèmes. Il est donc évident que l'analyse de tâche que nous avons simulée en laboratoire a conduit à un recensement des besoins informationnels sans doute non conforme à la réalité. Notons que lors de cette validation, 55 vues graphiques ont été interactivement générées par le concepteur (un exemple de vue en cours de création est visible en figure 2.6).<sup>12</sup>

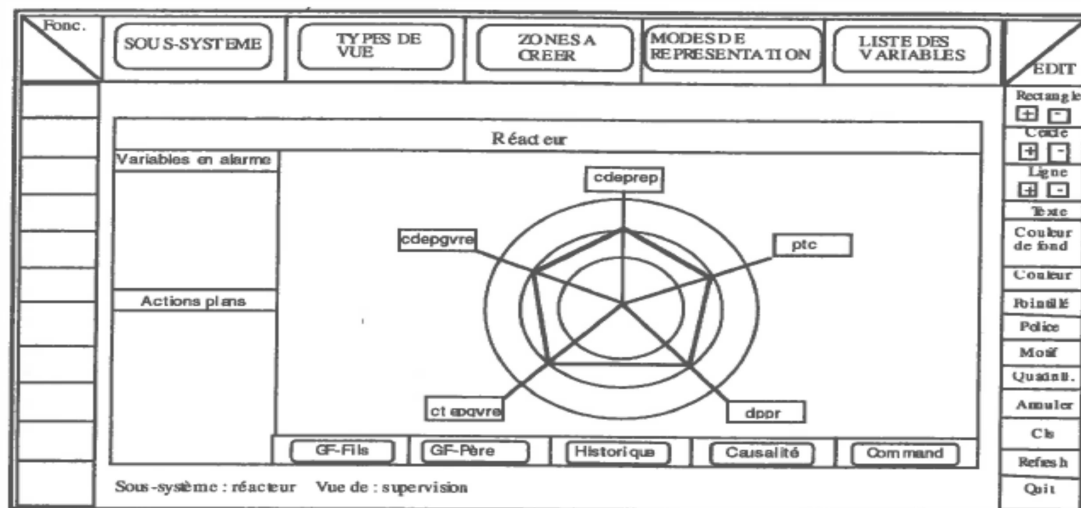


Figure 2.6 : Vue graphique en cours de création interactive

Cette première validation a prouvé la validité de la démarche : à partir d'une description de l'application, il est possible de générer automatiquement des spécifications, et ensuite de les mettre à la disposition du développeur afin de le guider dans la réalisation des vues. Il s'agit maintenant d'effectuer d'autres expériences à partir de cas réels et d'améliorer en conséquence chaque module.

#### 2.1.2.4. Conclusion sur l'approche

ERGO-CONCEPTOR est pour l'instant un outil de laboratoire, dont chacun des modules peut être maintenant amélioré. Par exemple, suite à une évaluation menée avec des utilisateurs potentiels (MILLET, 1993), il s'avère que la description par page-écran textuelle est fastidieuse et source d'interrogations et d'oublis, rendant nécessaire dans la prochaine version une description graphique et plus réaliste. Toutefois, les résultats de la première validation technique sont très positifs dans la mesure où la génération interactive d'imagerie selon cette démarche est d'ores et déjà possible.

Malgré ses faiblesses actuelles, la démarche suivie par ERGO-CONCEPTOR est prometteuse à plus d'un titre. En effet, quand on se place dans un domaine d'application où il existe déjà des conventions graphiques validées par des experts en communication homme-machine et utilisées par les développeurs, la génération de vues peut devenir un travail routinier. Ceci est d'autant plus vrai si le développeur est amené à réaliser des bibliothèques de vues de plusieurs dizaines voire de plusieurs centaines de vues graphiques où il n'est fait que très rarement appel à l'imagination ou à la créativité. En effet, et sans discuter de l'adéquation des vues aux besoins informationnels pour les différentes tâches à réaliser, on retrouve le plus souvent, dans ces vues, un nombre limité de modes de représentation, de symboles et de codages colorés ayant fait leurs preuves.

<sup>12</sup> Ce "concepteur" était en fait un des concepteurs d'ERGO-CONCEPTOR. Il connaissait donc parfaitement les points forts et les points faibles du système et a su l'utiliser au maximum de ses possibilités. Il s'agira maintenant de mettre ERGO-CONCEPTOR entre les mains de concepteurs industriels, ce qui nous permettra d'améliorer l'ergonomie d'utilisation des trois modules.

De plus, dans certains cas, suite à des modifications apportées aux installations, la mise à jour par le développeur de plusieurs vues peut s'avérer indispensable. Toutefois, des risques d'erreurs de sa part existent. Il apparaît ainsi à ce niveau l'utilité d'un module de description. En effet, le travail nécessaire peut se ramener à des modifications ponctuelles portées au moyen de celui-ci. Les routines du module de génération des spécifications se chargeront alors de compiler la nouvelle base de données et généreront un nouveau fichier de spécifications prenant en considération les modifications à répercuter sur l'ensemble des vues. Cette technique semble être intéressante du fait qu'elle peut apporter un gain de temps et surtout une certaine rigueur dans les modifications des vues concernées (MOUSSA et KOLSKI, 1992).

Cette approche originale d'ingénierie, considérant explicitement des facteurs techniques et humains, laisse présager des outils d'ingénierie qui seront confiés aux développeurs d'imagerie à moyen ou long terme.

### 2.1.3. Prototypage et système de spécification graphique d'interface

Un de nos thèmes de recherche concerne le prototypage d'imagerie industrielle à l'aide de spécifications graphiques interactives (POULAIN, KOLSKI et MILLOT, 1991). Cette contribution est le fruit d'une collaboration entre la CSEE (Compagnie des Signaux et Equipements Electroniques), 3IP (Société pour l'Innovation, l'Informatique Industrielle et la Productique) et le LAIH dans le cadre d'un projet ANVAR. Le travail décrit a conduit à la thèse de Thierry POULAIN (1994).

Au cours de ce projet, un outil graphique de prototypage a été développé par le LAIH et intégré dans un atelier de génie logiciel, appelé ATLAS (ATeLier Logiciel pour l'Animation de Synoptiques). Cet atelier, destiné à la spécification, la réalisation et l'évaluation d'imageries embarquées, avait été prévu initialement pour des applications de la Marine Nationale. Ces applications concernent la supervision de systèmes complexes, à l'aide d'images symbolisant les différents composants de l'installation. L'animation revient à simuler graphiquement la propagation de matière ou d'énergie dans un ou plusieurs réseaux, à partir de données provenant de capteurs. L'atelier vise à faciliter à l'équipe de développement un travail coopératif exploitant au maximum les compétences de chaque intervenant par l'intermédiaire d'un outil commun (figure 2.7).

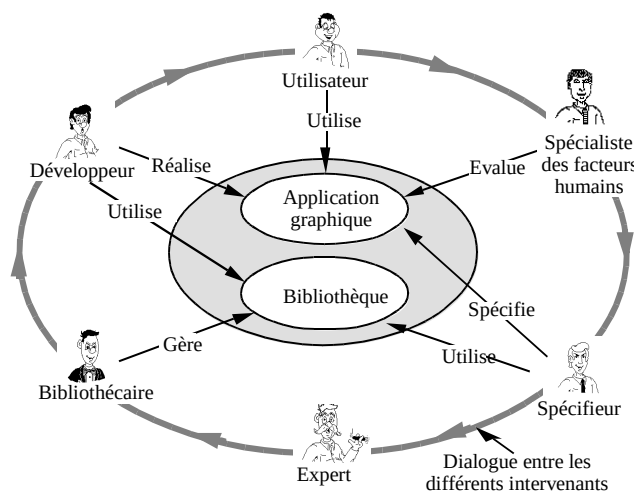


Figure 2.7 : Les différents intervenants (POULAIN, 1994)

Ainsi, le spécifieur a pour fonction de spécifier l'application qui sera réalisée (POULAIN et KOLSKI, 1992). Dans ce but, il fait intervenir l'expert de l'installation qui a de sa structure et de son fonctionnement une connaissance profonde. La notion de bibliothécaire a également été introduite : son rôle est de gérer une bibliothèque de composants graphiques de base à partir de laquelle d'une part le spécifieur

peut décrire l'installation et d'autre part le développeur peut concevoir l'application. En effet le développeur a pour fonction de produire le logiciel qui sera effectivement implanté sur le site. Le développement fait également intervenir d'une part l'utilisateur, et d'autre part le spécialiste des facteurs humains qui garantit la prise en compte de ceux-ci à certaines étapes de la démarche.

Cette section se compose de trois parties. La première positionne notre contribution dans l'atelier ATLAS. La seconde s'attarde sur le principe de spécification graphique. La troisième présente les résultats obtenus pour la première validation technique de l'outil de prototypage.

### ***2.1.3.1. Intégration d'un outil de prototypage dans un atelier logiciel***

Dans un premier temps, la structure matérielle d'ATLAS est précisée. Puis, les modules assurant la mise en œuvre du prototypage sont positionnés dans l'atelier. Enfin, les fonctionnalités principales permettant de spécifier graphiquement et de réaliser un prototype sont expliquées.

#### ***2.1.3.1.1. Structure matérielle d'ATLAS et logiciels de base***

L'atelier est scindé en : (i) une machine de développement SUN/SPARC qui sert de support à l'atelier pour les aspects de spécification, de réalisation des synoptiques et de validation technique, (ii) une machine cible, appelée Unité de Traitement Vidéo (UTV), qui exécute le code résultant de la phase de développement. Ce code est téléchargé via le réseau Ethernet de la SUN vers l'UTV. Cette configuration a été retenue dans la mesure où les applications visées sont implantées dans du matériel dit "endurci" répondant à des normes de fonctionnement particulières. Dans cette optique, une machine cible spécifique a été réalisée pour répondre aux critères du domaine d'application propre à la Marine Nationale.

Ainsi, une application sera engendrée sur la machine de développement, pour être ensuite transférée sur la machine cible, qui sera installée sur le site. C'est donc sur la machine cible que seront assurées les fonctions d'acquisition et de filtrage en ligne de données, de détermination des états de composants instrumentés ou non, de propagation d'états logiques dans le réseau, de détection d'incohérence, et d'animation graphique et de dialogue avec les utilisateurs. Sur la machine cible, ces fonctions se concrétisent par des processus communiquant entre eux par des tables de données. Ces processus sont construits autour du système d'exploitation VxWorks qui intègre un noyau temps réel.

L'atelier est construit autour des outils logiciels suivants : (i) TeleUse (de TeleLogic Europe), générateur d'interfaces sous MOTIF ayant permis de décrire l'interface homme-machine d'utilisation de l'atelier sur la machine de développement SUN, (ii) DATAVIEWS, utilisé pour mettre en œuvre les fonctionnalités principales de prototypage à savoir la description des installations et des synoptiques sur la machine de développement, mais également pour leur animation graphique sur la SUN et sur la machine cible, (iii) ORACLE, servant au stockage des données décrivant l'application.

#### ***2.1.3.1.2. Positionnement des fonctionnalités de prototypage dans l'atelier***

L'atelier ATLAS est doté d'un ensemble de fonctionnalités accessibles au travers de trois interfaces distinctes, l'une destinée au bibliothécaire, la seconde au développeur et la troisième au spécifieur (voir figure 2.8). L'atelier repose sur une bibliothèque de composants symbolisant des pompes, des vannes, des moteurs par exemple, gérés par le bibliothécaire.

A partir des éléments de cette bibliothèque et à l'aide des fonctionnalités de prototypage, le spécifieur assisté de l'expert décrit interactivement le réseau de l'installation à superviser, constitué

particulièrement de différents composants connectés entre eux. Puis, il réalise un ou plusieurs prototypes des images. Celles-ci peuvent être ensuite évaluées en dynamique à l'aide d'un simulateur sur SUN, avec les utilisateurs, et en liaison avec le spécialiste des facteurs humains, et ceci afin d'évaluer l'adéquation de l'imagerie aux besoins informationnels.

Puis, un module de récupération permet au développeur d'extraire une partie de la base de données "prototype" ainsi que les images validées, et ceci afin de constituer les images finales et un sous-ensemble de la base de données nécessaire à l'animation des images sur la machine cible. Il complète ensuite les informations nécessaires à la production du code qui sera ensuite téléchargé puis exécuté sur la machine cible. Un second simulateur, appelé "simulateur cible", ayant les mêmes fonctionnalités que le simulateur sur SUN, a pour rôle de valider, non pas le contenu des images, mais des aspects techniques, par exemple des adresses pour l'acquisition des données.

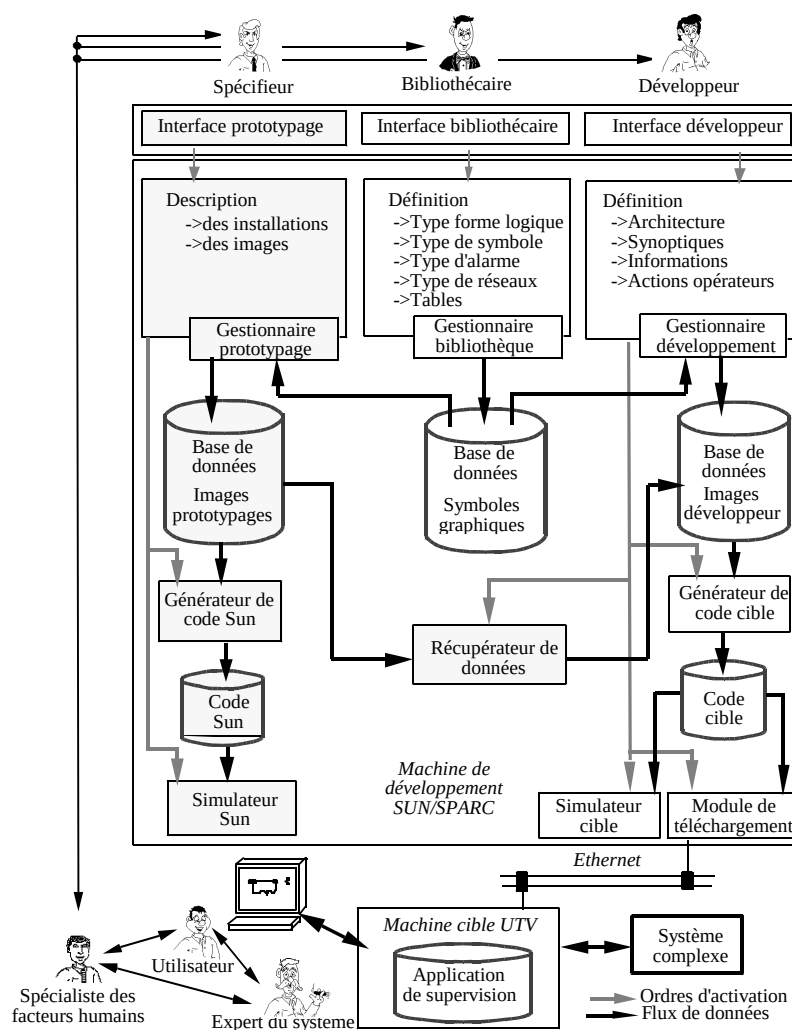


Figure 2.8 : Intégration de fonctionnalités de prototypage dans l'atelier ATLAS (POULAIN, GERME et KOLSKI, 1993)

Comme on le voit sur la figure, et afin d'assurer les fonctionnalités de prototypage, quatre modules ont donc été prévus :

- *Gestionnaire de prototype* : il permet au spécifieur de se décharger au maximum de tâches répétitives de description. Ses fonctions lui permettent de décrire graphiquement le ou les réseaux à superviser. Puis il lui est possible à partir de cette description de créer rapidement un prototype de l'imagerie, qu'il conviendra ensuite d'évaluer.

- *Récupérateur de données* : ce module a pour but de recopier une partie des informations de la base de données de prototypage dans la base de données de développement.
- *Générateur de code Sun* : ce générateur de code présente les mêmes fonctionnalités que le générateur de code cible, à la différence près que le code est exécutable sur la Sun.
- *Simulateur Sun* : il remplit les mêmes fonctionnalités que celui dédié à la cible. Néanmoins dans ce cas l'évaluation s'effectue directement sur la machine de développement.

Notons que parmi les quatre derniers modules supportant le prototypage, seul le gestionnaire de prototypage a fait l'objet de développements de la part du LAIH. Les trois autres modules sont actuellement en cours de conception par 3IP et la CSEE et devraient être opérationnels fin 1994. Basé sur des spécifications graphiques, le gestionnaire de prototypage qui, en plus de l'aspect méthodologique, a fait l'objet de la contribution du LAIH à ce projet, est décrit maintenant.

### **2.1.3.2. Description du gestionnaire de prototypage**

Rappelons que c'est le spécifieur qui utilise le gestionnaire de prototypage, prenant la forme d'un éditeur dédié offrant deux fonctionnalités principales : de description des installations et de description des images (POULAIN, 1994 ; VILAIN, POULAIN et KOLSKI, sous presse).

#### **2.1.3.2.1. La description des installations**

Cette première fonctionnalité permet au spécifieur de décrire graphiquement les installations selon la méthode d'analyse systémique "diagramme blocs". Elle consiste à partir du système global pour arriver aux composants élémentaires par affinements successifs. Elle présente l'avantage d'aboutir à une description réaliste et facilite également le dialogue entre les différents intervenants. Afin de faciliter au spécifieur cette description, un ensemble de fonctionnalités accessibles par trois menus distincts est mis à sa disposition :

- Des fonctions d'édition permettent, d'une part, de créer, de supprimer et de manipuler des composants élémentaires (pompes, vannes, disjoncteur...), et d'autre part, de créer différents sous-systèmes et de naviguer dans ceux-ci. Il est possible également de gérer les liaisons entre composants et/ou sous-systèmes.
- Des outils permettent au spécifieur de couper et de coller des parties des installations décrites, d'effectuer différents zooms, ou de visualiser les caractéristiques d'un composant.
- Un menu de commande gère les fichiers résultant de la description. Celles-ci permettent de recharger une description, de sauvegarder, de fermer sans sauvegarder, ou encore de quitter le gestionnaire de prototypage.

Tout au long de sa description, le réseau des constituants est mémorisé dans la base de données de prototypage (figure 2.8) qui sert par la suite pour décrire le synoptique, de même que pour simuler la propagation de matière ou d'énergie dans le réseau.

#### **2.1.3.2.2. La description des images**

L'interface de prototypage permet ensuite au spécifieur de décrire le synoptique. Pour composer la partie des images appelée "synoptique", partie correspondant à l'image physique des installations, le



spécifieur copie sur les images de description des installations les parties qui l'intéressent pour les "coller" sur les images. Puis, il complète ses images par des informations complémentaires.

Ce principe de description a été choisi car il respecte les méthodes de travail des développeurs des applications marines visées. Ce point a été mis en évidence lors de l'étude de leurs besoins : dans la majorité des cas, ceux-ci transposent directement sur les images des parties issues de la description des installations (POULAIN, 1994). Ce principe de description est résumé en figure 2.9. Comme pour la description du synoptique, un ensemble de fonctionnalités est accessible :

- Des fonctions d'édition permettent de manipuler des objets sur l'écran, d'en créer, d'en effacer, de dessiner le fond de plan de l'image, de naviguer dans la description des installations afin d'en copier des parties, etc.
- Des outils permettent de couper des éléments, d'en copier, d'en coller sur les images, d'afficher des informations relatives à un composant sélectionné, etc.
- Un menu de commandes est utilisé pour gérer les fichiers associés aux différentes vues de prototype. Cette gestion comprend entre autre la sauvegarde et le chargement des vues.

Lorsque le prototype de l'imagerie est terminé, il importe de le soumettre aux utilisateurs et d'en effectuer une évaluation en collaboration avec l'ergonome. Les vues validées seront ensuite prises en charge par le développeur qui complètera les informations manquantes et générera le code cible.

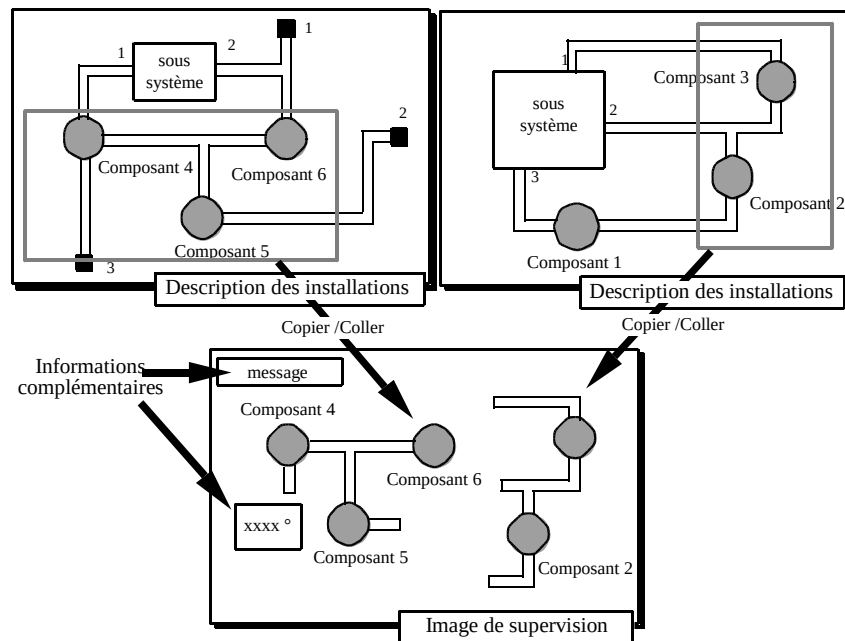


Figure 2.9 : Principe de description des images (POULAIN, 1994)

#### 2.1.3.2.3. Conclusion sur le gestionnaire de prototypage

Ce gestionnaire est terminé et a été intégré dans l'atelier ATLAS. Il permet à un spécifieur non-informaticien, et ceci selon des concepts simples, la description graphique de l'ensemble des systèmes, sous-systèmes et composants du ou des réseaux à superviser. Cette description fait office de spécification graphique servant de base à la génération de l'imagerie, ou du moins d'un prototype de celle-ci. En effet, pour créer des images, il "suffit" de récupérer à l'aide de la souris des parties de la description précédente, et d'ajouter éventuellement des informations complémentaires.

Mais, pour que le prototypage soit vraiment effectif, il importe que les trois modules permettant respectivement la récupération des données issues du prototypage, la génération de code sur SUN et la simulation sur SUN, soient également intégrés à l'atelier ATLAS. Il sera alors possible d'effectuer une évaluation complète de l'utilisation de l'atelier. Pour l'instant, seule une validation technique du gestionnaire a été effectuée.

### 2.1.3.3. Validation technique du gestionnaire de prototypage

Une validation technique de l'ensemble des fonctionnalités du gestionnaire de prototypage a été réalisée en laboratoire par POULAIN (1994). Celui-ci est parti d'une installation de type centrale nucléaire. Ce choix se justifie par le fait qu'il avait déjà travaillé sur cette installation dans le cadre du projet ALLIANCE, cité en 2.1.1.3. Pour cette validation, l'installation a bien entendu été simplifiée.

A l'aide des fonctionnalités de description des installations, celles-ci ont d'abord été décrites au niveau le plus haut à l'aide de trois sous-systèmes principaux : (i) le circuit primaire correspondant à la boucle de chauffage sodium, (ii) le circuit de vidange et remplissage sodium et (iii) le circuit secondaire correspondant à la boucle du générateur de vapeur. Chacun de ces sous-systèmes a ensuite été décrit à l'aide de la bibliothèque de composants représentant des vannes, des pompes, de réchauffeurs, des échangeurs, etc. Cette description a ensuite été sauvegardée, et a servi de support pour la réalisation des images.

Ainsi, à l'aide des fonctionnalités de description du synoptique, plusieurs images ont été réalisées : principalement une vue de supervision synthétisant les installations, une seconde détaillant le circuit primaire, et une troisième détaillant le circuit secondaire. Il a été possible rapidement de récupérer sur la description des installations des parties, de les coller sur les images, et de compléter celles-ci par des informations complémentaires comme on le voit sur la figure 2.10 : sur l'image en cours de réalisation représentant le circuit secondaire, on distingue un barre-graphe indiquant la température du condenseur.

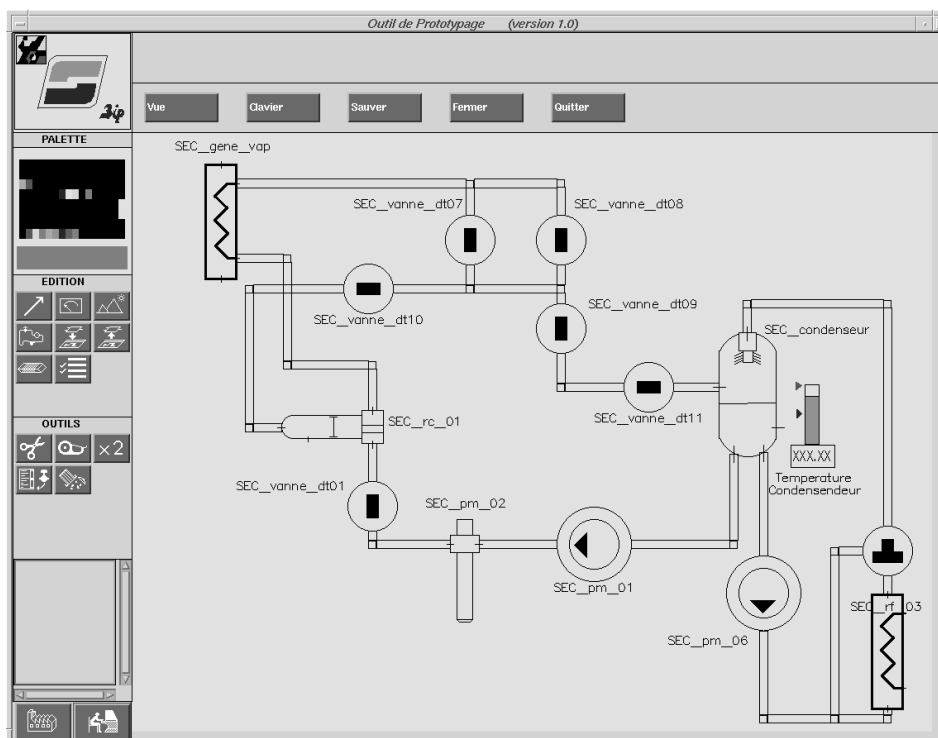


Figure 2.10 : Image en cours de réalisation

Notons que lors de la description, plusieurs composants n'existaient pas en bibliothèque. De ce fait, celle-ci a été complétée à l'aide de l'interface bibliothécaire : la modélisation d'un composant a consisté à définir les comportements graphique et logique. Globalement, lors de cette première validation, le gestionnaire s'est avéré pratique d'utilisation. Il faut maintenant le mettre entre les mains d'utilisateurs pour l'évaluer plus en profondeur.

#### **2.1.3.4. Conclusion**

Dans la démarche de prototypage proposée par l'atelier, la spécification de l'imagerie est graphique et s'adresse à un spécifieur non-informaticien. Cette manière de spécifier l'imagerie est originale dans l'industrie. Elle a été proposée et mise en oeuvre lors d'une collaboration industrielle entre CSEE, 3IP et le LAIH dans le cadre d'un projet ANVAR, pour déboucher sur un gestionnaire de prototypage intégré dans l'atelier ATLAS. Ce travail a fait l'objet de plusieurs rapports de contrat et d'un manuel utilisateur (POULAIN, VILAIN, KOLSKI et MILLOT, 1991, 1992 et 1993). Ce gestionnaire offre deux fonctionnalités principales au spécifieur : la description graphique des installations et la description du synoptique.

Une première version de l'atelier est actuellement disponible, et utilisée par la Marine Nationale. Elle permet le développement des synoptiques, mais elle n'est pas encore dotée des fonctionnalités de prototypage. Le développeur doit donc créer pour l'instant complètement les bases de données à l'aide de masques de saisie Oracle, ainsi que les images grâce à l'éditeur DVdraw de DATAVIEWS. La plupart des fonctionnalités de prototypage devraient toutefois être intégrées dans la version finale et commerciale de l'atelier prévue pour fin 1994.

#### **2.1.4. Approche d'interface "intelligente"**

Nous avons contribué au domaine des interfaces dites "intelligentes", par l'étude et la mise en oeuvre d'une approche originale appelée Module Décisionnel d'Imagerie (KOLSKI, TENDJAOUI et MILLOT, 1990). Cette recherche a fait l'objet de la thèse de doctorat de Mustapha TENDJAOUI (1992).

Le Module Décisionnel d'Imagerie (M.D.I.) prend la forme d'un système "intelligent" de gestion de l'imagerie en dynamique qui s'intègre dans une chaîne logicielle de supervision. Une des originalités du M.D.I. consiste en la formalisation de trois concepts dans l'interface : QUOI (quelles informations) présenter à l'utilisateur, QUAND lui présenter l'information et COMMENT la lui présenter.

Afin de présenter cette contribution, cette section est composée de quatre parties. Dans la première, le M.D.I. est localisé dans le système homme-machine et ses concepts de base sont définis. La deuxième décrit une plate-forme logicielle expérimentale intégrant une interface "intelligente" de type M.D.I. La troisième partie s'attarde sur la réalisation du M.D.I. en décrivant le coeur de celui-ci prenant la forme d'un système à base de connaissance. Enfin, la quatrième présente les résultats d'une première validation en laboratoire ayant pour but de tester la validité des décisions prises par le Module Décisionnel d'Imagerie.

##### **2.1.4.1. Localisation et Principes du M.D.I.**

La localisation du M.D.I. est visible en figure 2.11 (TENDJAOUI, KOLSKI et MILLOT, 1990) : le calculateur centralise l'ensemble des données instrumentées du système complexe. Ces données sont accessibles au M.D.I. et aux modules d'assistance. A partir de ces données, les modules d'assistance déduisent des informations telles que des prédictions, des diagnostics ou des procédures de reprise. Ces informations sont alors transmises au M.D.I. qui sélectionne celles qui doivent être présentées à l'utilisateur, et ceci selon une présentation la plus ergonomique possible. Le M.D.I. est lié aux modules d'assistance par

les données et les informations qu'il doit gérer, mais il est aussi un composant indépendant de par la nature particulière des règles contenues dans ses bases (TENDJAOUI, KOLSKI, GAMBIEZ et MILLOT, 1991).

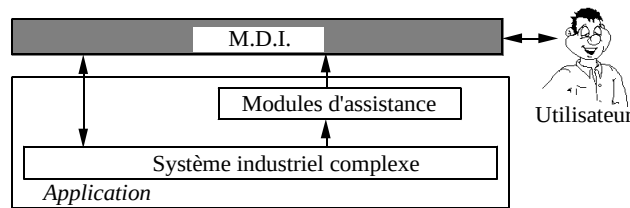


Figure 2.11 : Localisation du M.D.I.

Le M.D.I. vise à être autonome et capable de s'adapter aux besoins des utilisateurs. Il doit donc intégrer des connaissances sur : (i) les différents contextes opératoires du système à superviser, (ii) les utilisateurs de l'interface, (iii) leurs tâches à réaliser. Ces connaissances doivent permettre de répondre aux trois problèmes suivants, figure 2.12 (TENDJAOUI, KOLSKI et MILLOT, 1991a) :

- **QUOI** présenter à l'utilisateur, c'est à dire : quelles sont les informations pertinentes et utiles à afficher compte tenu des limites de perception humaines et aussi de la taille réduite des écrans. Nous considérons que le "QUOI" intègre le "POURQUOI", par l'utilisation de niveaux de justification et d'explication de l'information présentée.
- **QUAND** lui présenter les informations sélectionnées, notamment en tenant compte du contexte opérationnel.
- **COMMENT** les lui présenter, c'est à dire : quels sont les modes de représentation les mieux adaptés du point de vue ergonomique.

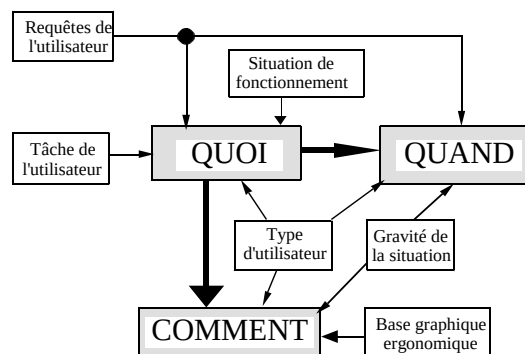


Figure 2.12 : Les Concepts "QUOI / QUAND / COMMENT" du M.D.I.

Les concepts du "QUOI", du "QUAND" et du "COMMENT" sont liés entre eux par l'intermédiaire d'autres caractéristiques du système homme-machine : situation, gravité, requêtes de l'utilisateur, etc. Il est donc nécessaire que l'architecture logicielle de l'interface soit bâtie autour de l'ensemble de ces notions (KOLSKI, TENDJAOUI et MILLOT, 1992). Après avoir présenté les objectifs et concepts de base du M.D.I., nous proposerons par la suite une architecture possible. Au vu de toutes les approches d'interfaces "intelligentes" qui ont été décrites dans l'état de l'art, il est bien évident que d'autres architectures peuvent être envisagées. Celle que nous proposons a pour avantage de considérer *explicitement* les 3 concepts "QUOI", "QUAND" et "COMMENT" sous la forme de bases de connaissances.

#### 2.1.4.2. Réalisation d'une plate-forme expérimentale

Une plate-forme expérimentale intégrant une interface "intelligente" de type M.D.I. a été développée au laboratoire sur ordinateur VAX sous VMS dans la cadre de la thèse de TENDJAOUI (1992). Elle est composée d'un ensemble de modules informatiques partageant une zone mémoire commune, figure 2.13 (TENDJAOUI, KOLSKI et MILLOT, 1991b). On distingue principalement :

- *Un simulateur* : il simule le fonctionnement simplifié du circuit de refroidissement d'une centrale, en étant constitué d'un circuit primaire et un circuit secondaire. Il correspond en fait à un modèle qualitatif simple, composé de nœuds reliés entre eux par des arcs. Si une perturbation arrive sur un des nœuds, elle est propagée dans les arcs qui partent du nœud.
- *Quatre modules d'assistance* : (i) un module de prédiction correspondant au modèle qualitatif précédent accéléré, (ii) un module de traitement des alarmes, (iii) un générateur de plans d'actions, un plan étant du type "*Pour corriger la variable X il faut agir sur la variable Y de Z dans le sens (-,+) avant T secondes*" et (iv) un générateur de justifications de plans d'actions. Les informations qu'ils génèrent ne sont pas affichées automatiquement, mais sont stockées en zone mémoire commune. Leur affichage est ensuite contrôlé par le M.D.I.<sup>13</sup>
- *Un superviseur* : celui-ci gère la coordination et les communications à travers la zone mémoire commune. Il ne sera pas décrit dans la mesure où il se contente d'assurer les échanges de données.
- *Un module de gestion des scénarios de pannes* : dans un but expérimental, chaque scénario peut être décrit à l'aide d'un éditeur spécialisé.
- *Un module de gestion des actions et des requêtes de l'utilisateur* : ce module sauvegarde après interprétation celles-ci dans des fichiers. Pour les mêmes raisons que pour le superviseur, une description détaillée de ce module de gestion n'est pas indispensable.

---

<sup>13</sup> Nous avons modélisé les justifications de chaque défaut et de son plan d'action par deux réseaux de variables. Le premier, appelé "justification de surface", présente la variable sur laquelle l'alarme s'est déclenchée ainsi que l'ensemble des causes possibles, permettant ainsi de limiter le champ de recherche de la cause du défaut. Le deuxième réseau, appelé "justification profonde", représente la variable sur laquelle l'alarme s'est déclenchée et les causes les plus probables auxquelles elle est reliée ainsi que les variables intermédiaires perturbées.

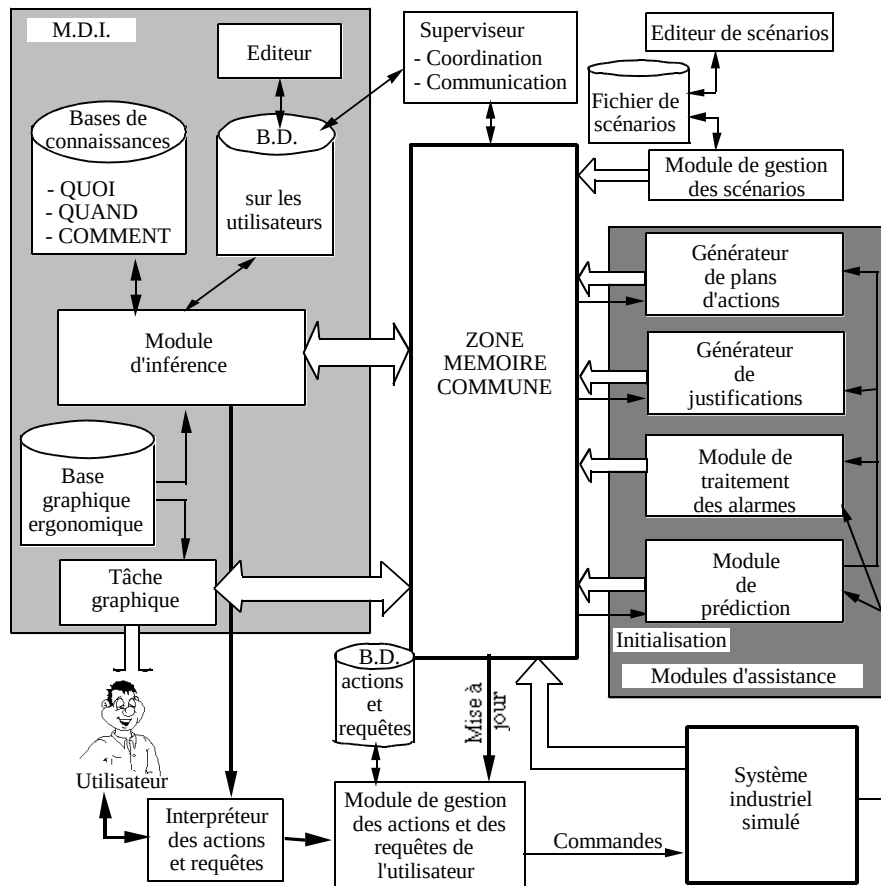


Figure 2.13 : Architecture de la plate-forme expérimentale

Le M.D.I., et plus particulièrement l'aspect de gestion "intelligente" de l'imagerie, fait l'objet du paragraphe suivant .

#### 2.1.4.3. Le gestionnaire "intelligent" de l'imagerie, cœur du M.D.I.

Le M.D.I. intègre des connaissances répondant aux trois problèmes "QUOI", "QUAND" et "COMMENT", ainsi qu'un moteur d'inférence chargé d'exploiter les règles contenues dans les bases. Il doit en fait pouvoir composer et enchaîner "intelligemment" les vues en fonction du contexte, des tâches de l'utilisateur, de son niveau d'expérience, de ses habitudes, de la situation de fonctionnement, etc. En résumé, il doit définir les décisions à prendre pour répondre aux questions de "QUOI" présenter et "QUAND" et "COMMENT" présenter l'information.

##### 2.1.4.3.1. Principes utilisés pour la composition et l'enchaînement des vues

L'utilisateur dispose : (i) d'une fenêtre principale, (ii) d'une fenêtre présentant des informations importantes telles que les alarmes et les conseils de correction, (iii) d'une fenêtre destinée aux commandes, (iv) d'une fenêtre contenant les commandes d'appel de vues (requêtes de l'utilisateur) et (v) d'une fenêtre affichant des affichages secondaires que le M.D.I. propose sans qu'elles soient demandées par l'utilisateur.

Dans le premier cas où l'utilisateur effectue une requête pour l'affichage d'une information, le M.D.I. répond à celle-ci, car l'utilisateur doit pouvoir à tout moment accéder à l'ensemble des vues. Dans le second cas où l'affichage est contrôlé par le M.D.I., l'affichage est fonction des connaissances centralisées dans les bases "QUOI", "QUAND" et "COMMENT". En effet, le M.D.I. évalue la situation et décide si une vue doit être affichée dans la fenêtre principale, si l'utilisateur la demande ou dans la fenêtre secondaire si le M.D.I.

le décide. En effet, si la situation se dégrade, des justifications peuvent être affichées dans la fenêtre secondaire permettant à l'utilisateur de vérifier le plus rapidement possible les conseils d'actions qui auront été déjà proposés.

Pour l'ensemble des aspects graphiques gérés ou non par le M.D.I., il a été nécessaire de réaliser une bibliothèque graphique supposée répondre a priori à des critères ergonomiques de présentation d'information sur écran. Cette bibliothèque est pilotée par le gestionnaire "intelligent" de l'imagerie, décrit maintenant.<sup>14</sup>

#### 2.1.4.3.2. Le gestionnaire "intelligent" d'imagerie

Le gestionnaire "intelligent" de l'imagerie consiste en un système d'inférence capable de traiter et d'exploiter l'ensemble des connaissances relatives au "QUOI", au "QUAND" et au "COMMENT". Il comporte un moteur d'inférence spécifique développé à l'aide du langage C, une base de connaissance sur le "QUOI", une sur le "QUAND", et une autre sur le "COMMENT". La figure 2.14 explicite les entrées et les sorties d'informations utilisées par le moteur d'inférence.

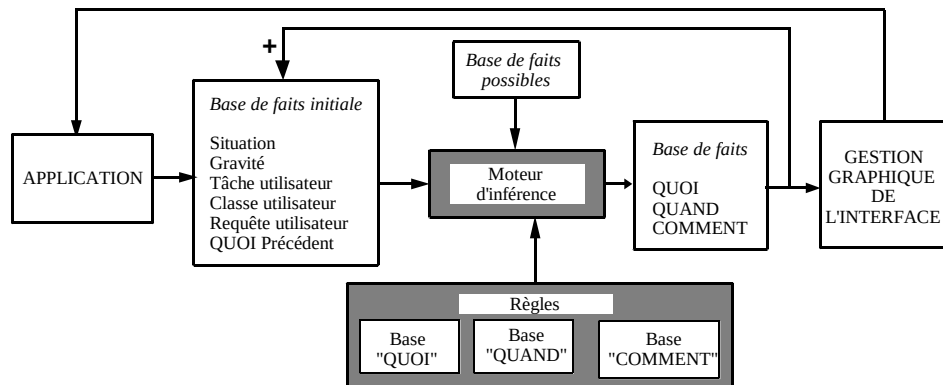


Figure 2.14 : Schéma de principe du cœur du M.D.I.  
(LE STRUGEON, TENDJAOUI et KOLSKI, 1992)

Il fonctionne en chaînage avant car on cherche à obtenir le plus de faits possibles : tous les faits déductibles sont déduits. Des faits multivalués représentent : (i) l'état de fonctionnement par les faits "Situation", "Gravité"<sup>15</sup> et "Tâche utilisateur", (ii) le type d'utilisateur et ses requêtes éventuelles par "Classe utilisateur" et "Requête utilisateur", (iii) l'état précédent de l'interface par "QUOI précédent". Ces faits font partie de la base de faits initiale. Les données nécessaires à la constitution de cette base sont fournies au système expert par un superviseur. Le moteur d'inférence utilise cette base de connaissance pour déduire de nouveaux faits : les faits QUOI, QUAND et COMMENT, qui correspondent à ce qu'il faut afficher, quand et comment l'afficher. La manière dont les bases sont créées sera décrite en 2.2.2. Deux exemples simplifiés de règles issues des bases "QUOI" et "QUAND" sont visibles ci-dessous :

<sup>14</sup> Les différents modes de représentation et les attributs graphiques qui ont été spécifiés puis réalisés doivent répondre directement à la question "COMMENT présenter l'information". Dans notre cas, nous avons considéré pour l'instant quatre types de vues : (i) de supervision, (ii) de justification peu détaillée, (iii) de justification très détaillée et (iv) d'affichage de courbes. Des exemples de vues sont fournis dans la thèse de TENDJAOUI (1992).

<sup>15</sup> La gravité est définie grâce à trois paramètres traduisant l'état de fonctionnement du système (i) *le degré d'insécurité* traduisant une échelle de gravité du système, prenant en compte de manière explicite des consignes de sécurité, (ii) *le degré d'anormalité de la production*, traduisant une échelle du taux de production en énergie de la centrale, (iii) *le degré d'urgence*, exprimant le délai imparti pour effectuer une compensation d'un défaut éventuel. Grâce à cette classification, des classes d'alarmes possibles sont identifiées. Chaque classe est associée à un degré de gravité de la situation. Les niveaux de gravité sont calculés par la fonction suivante :  $Degré(gravité) := Degré(insécurité) + Degré(production) + Degré(urgence)$

La classification permet d'identifier cinq situations de fonctionnement possibles : optimale, normale, anormale, critique et irrécupérable. La formulation du degré de gravité, comme une agrégation de trois critères a été choisie volontairement simple dans notre étude, car l'objectif est ici de tester la faisabilité des concepts du M.D.I.

SI	gravité	=	5	
ET SI	classe-utilisateur	=	expert	
ET SI	situation	=	critique	
ET SI	requête-utilisateur	=	aucune	
ALORS	QUOI	=		Justification-profonde
SI	gravité	=	1	
ET SI	classe-utilisateur	=	expert	
ET SI	requête-utilisateur	=	aucune	
ET SI	QUOI	=	plan d'action	
ALORS	QUAND	=		Lorsque-requête

La première règle exprime que si la gravité est de 5, que l'utilisateur est expert, que la situation est critique et qu'aucune requête n'a été effectuée, alors il s'agit de préparer une justification profonde de la cause du défaut. La seconde règle exprime que si la gravité n'est que de 1, que l'utilisateur est expert, qu'il n'a encore fait aucune requête et qu'un plan d'action est disponible, alors celui-ci ne doit être affiché qu'à la demande de l'utilisateur. Celui-ci a donc le loisir de définir sa propre stratégie, afin éventuellement par la suite de la confronter à celle issue des différents modules d'assistance.

#### ***2.1.4.4. Première validation technique de la plate-forme***

Afin de valider techniquement la plate-forme intégrant le M.D.I., un premier scénario de pannes a été établi (KOLSKI, LE STRUGEON et TENDJAOUI, 1993). Ce scénario se compose de quatre défauts, à chaque fois espacés dans le temps de trois minutes, consistant en des croissances ou décroissances de variables représentatives du bon fonctionnement du système industriel simulé, ayant des répercussions sur d'autres variables. Pour ce scénario, un premier sujet a utilisé la plate-forme, selon un mode particulier précisé ci-après.

Dès que l'ensemble des modules est lancé, les variables du système simulé sont mises à leur valeur de consigne de sorte que la situation de fonctionnement soit optimale. Trois minutes plus tard, le premier défaut apparaît. Les données enregistrées lors de cette pré-expérimentation étaient : le niveau de gravité engendré par la situation de fonctionnement à un instant donné, le type de vue affichée dans la fenêtre principale, le type de vue affichée dans la fenêtre secondaire, et les actions du sujet. Dans le but de pouvoir identifier son activité, et ceci tout au long de l'expérimentation, nous lui avons demandé de sélectionner certains objets graphiques en les visant à l'écran avec la souris. De cette manière, il était possible à chaque instant d'identifier et de mémoriser la détection d'une alarme (si le sujet visait l'objet texte correspondant à l'alarme), la lecture d'un plan d'action, la recherche des causes possibles d'un défaut, la correction d'un défaut ou encore le suivi d'une action de correction.

L'évolution dans le temps des paramètres recueillis lors de la pré-expérimentation a ensuite été représentée graphiquement. On y retrouve : (i) les décisions du M.D.I. représentées par la gestion de l'affichage des alarmes et des plans d'actions, ainsi que par le contenu de la fenêtre secondaire, (ii) les réponses aux requêtes du sujet représentées par le contenu de la fenêtre principale, (iii) l'activité du sujet, (iv) l'évolution de la gravité de la situation (évaluée en permanence par le M.D.I.) engendrée par le fonctionnement du système simulé tout au long de l'expérience. Le lecteur se référera à la thèse de TENDJAOUI (1992) dans laquelle toutes les décisions prises par le M.D.I., au fur et à mesure des vingt minutes de cette pré-expérimentation, sont expliquées.

Au vu des données recueillies, la plate-forme expérimentale mise en œuvre est globalement opérationnelle. Il est maintenant possible d'entreprendre l'étape suivante consistant à évaluer ergonomiquement le M.D.I.



#### 2.1.4.5. Conclusion sur l'approche

Les travaux sur le M.D.I. s'intègrent dans le domaine de l'assistance en salle de supervision. Notre contribution réside dans le développement d'une interface "intelligente" originale. Ainsi, nous proposons d'utiliser un système à base de connaissance pour gérer *explicitement* à l'aide de règles de production les interactions entre le système, l'utilisateur et les différents modules d'assistance dont il dispose, en répondant aux trois questions suivantes : (1) QUOI présenter à l'utilisateur, (2) QUAND le présenter et (3) COMMENT le présenter. Par la réponse à ces questions, notre approche vise à s'adapter, notamment, aux besoins de l'utilisateur.

Une plate-forme expérimentale intégrant un ensemble de modules d'assistance à l'utilisateur et une interface "intelligente" de type M.D.I. a été développé au Laboratoire. Le système complexe considéré est un simulateur du circuit de refroidissement simplifié d'une centrale. La plate-forme expérimentale est destinée à évaluer l'apport pour l'activité de l'utilisateur et pour le système homme-machine, d'un système de dialogue de ce type. Elle a été validée techniquement lors d'une première pré-expérimentation. De nombreuses autres expérimentations doivent maintenant être réalisées.

En parallèle à cette contribution aux interfaces dites "intelligentes", nous avons eu l'occasion de travailler sur certains des aspects graphiques utilisés par le M.D.I., lors d'un projet industriel en collaboration avec la société S2O (GAMBIEZ, TENDJAOU, KOLSKI et MILLOT, 1990). L'objectif de cette étude concernait la définition et la validation d'un module d'interface pour l'outil PREDEX de développement de système expert de filtrage et de gestion d'alarmes. Le but du module était de fournir une aide au développeur de l'imagerie selon la nature des informations fournies par PREDEX (prédiction de l'évolution du système, diagnostic, conseils d'action et leurs justifications), des informations issues du calculateur de surveillance et des commandes des utilisateurs. Ce module prenait la forme d'un générateur d'interface mis en oeuvre à l'aide de DATAVIEWS, paramétrable selon la nature du système, permettant d'assembler des objets graphiques prédéfinis dans une bibliothèque (GAMBIEZ, KOLSKI, TANG et MILLOT, 1990).

Notons qu'une réflexion menée avec René MANDIAU depuis 1991 a concerné l'utilisation potentielle des approches multi-agents pour la proposition d'architectures d'interfaces dites "intelligentes" (Cf. 1.3.2.5). Dans ce cadre, un premier article a porté sur une architecture d'interface "intelligente" à base d'agents (MANDIAU, KOLSKI et MILLOT, 1991). Un second article s'est penché sur la génération de plan dans une structure orientée agents (MANDIAU, KOLSKI, MILLOT et CHAIB-DRAA, 1991).

#### 2.1.5. Conclusion sur les systèmes mis en œuvre

C'est autour de quatre approches originales d'ingénierie des interfaces homme-machine que notre réflexion a été menée. Celles-ci nous ont permis d'aborder deux thèmes riches en concepts : la conception et l'évaluation d'interfaces. Dans ces thèmes, plusieurs orientations se sont dégagées : l'utilisation de bases de connaissance pour la conception et l'évaluation, la mise en place d'outils graphiques facilitant le prototypage, ou encore l'explicitation des concepts "QUOI", "QUAND" et "COMMENT présenter l'information" dans une approche d'interface dite "intelligente".

Lors de ces recherches, les connaissances utilisables pour la conception et l'évaluation ont pris progressivement énormément d'importance. En effet, en plus de recenser, structurer et formaliser ces connaissances dans les bases de SYNOP, ERGO-CONCEPTOR et du M.D.I., certaines d'entre elles ont été mises en forme dans des manuels et un didacticiel destinés à des développeurs d'imageries industrielles. C'est pourquoi la partie suivante est consacrée à l'étude et l'emploi de ces connaissances.

## **2.2. METHODES MISES EN PLACE DANS LE CADRE DE NOS RECHERCHES**

Dans la première partie du chapitre, nous avons donné un aperçu des systèmes réalisés au cours de nos recherches. Il s'agit maintenant de résumer plusieurs méthodes mises en place autour de ces systèmes.

D'abord, nous expliquons nos travaux relatifs au recensement de connaissances expertes. Ensuite, nous positionnons SYNOP et ERGO-CONCEPTOR dans la méthode en U. Puis, la méthode d'analyse mise au point durant le projet SOLVAY est décrite. Par la suite, nous présentons la méthode de création des bases du M.D.I. Enfin, nous terminons sur la méthode de développement d'imagerie axée sur le prototypage mise au point lors du projet ATLAS.

### **2.2.1. Recensement de connaissances expertes**

Lors des travaux relatifs au développement d'outils d'ingénierie d'interface homme-machine, nous avons recensé de nombreuses connaissances liés à la conception et l'évaluation d'interface homme-machine. Dans un premier temps, nous résumons la démarche ayant abouti aux connaissances centralisées dans les bases de SYNOP, ERGO-CONCEPTOR et du M.D.I.

Pour les applications industrielles complexes, les besoins des développeurs vis-à-vis de telles connaissances sont très importants. A ce sujet, le laboratoire a été sollicité à plusieurs reprises pour des projets de mise à disposition de développeurs de connaissances livresques et didactiques. C'est pourquoi nous présentons aussi le travail réalisé lors de collaborations de recherche avec GAZ DE FRANCE, B+ DEVELOPMENT et SOLVAY.

#### ***2.2.1.1. Recensement de connaissances pour les bases de SYNOP, ERGO-CONCEPTOR et du M.D.I.***

L'intégration de connaissances ergonomiques dans la base de SYNOP a constitué une étape essentielle dans sa réalisation. Mais ce travail a aussi largement influencé nos recherches suivantes, et particulièrement celles concernant la mise en place des bases d'ERGO-CONCEPTOR et du M.D.I. C'est pourquoi il nous paraît utile de résumer la démarche ayant permis d'aboutir aux bases de SYNOP. Celle-ci s'est inspirée de travaux menés en ingénierie de la connaissance dans l'équipe de Bernard HOURIEZ (1994). Elle se décompose en quatre étapes, figure 2.15.

L'identification du domaine a permis d'acquérir un niveau de compétence de dialogue avec des experts en communication homme-machine. Elle a commencé par une acquisition bibliographique de connaissance, aux sources multiples. Des interfaces développées en industrie ou en laboratoire ont également été étudiées, ce qui a permis de se familiariser avec les différents modes existants de présentation d'information. La deuxième étape a consisté en l'énoncé de concepts et de règles de présentation d'information et en la formalisation de ceux-ci en règles de production. Des concepts et règles issus de l'étape d'identification ont été proposés à des experts lors d'entretiens.

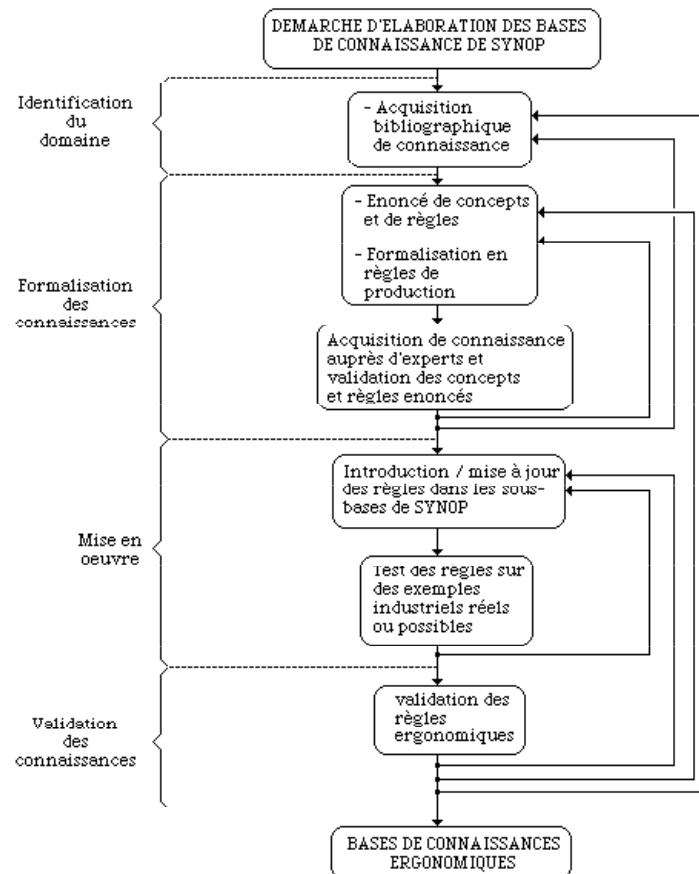


Figure 2.15 : Démarche d'élaboration des bases de connaissance de SYNOP

Plusieurs problèmes liés à l'acquisition de la connaissance ont alors surgi. Par exemple, la connaissance était répartie entre des spécialistes : chacun d'eux pouvait détenir une expertise partielle qu'il était souvent intéressant de regrouper. Ainsi, les experts cotoyés lors du projet SYNOP utilisaient des démarches de conception différentes. Il importait alors de trouver des compromis entre ces démarches. De plus, la connaissance ergonomique peut être très subjective et source variable d'interprétation. Elle était aussi souvent compilée suivant des concepts généraux qu'il fallait alors redécomposer en règles de production. Malgré ces contraintes, les entretiens permettaient la validation de ces concepts et règles, ainsi que l'énoncé d'autres concepts et règles. De plus, cette étape pouvait nous ramener à l'étape d'identification du domaine lorsque de nouveaux thèmes ergonomiques étaient abordés. Les concepts et règles formalisés en règles de production pouvaient ensuite être introduits par l'intermédiaire de l'interface expert dans les "sous-bases" de connaissance ou de métaconnaissance de SYNOP, lors d'une troisième phase de mise en oeuvre des connaissances. Puis, chaque règle était testée à partir d'exemples industriels réels ou simulés. Les résultats obtenus lors de la mise en oeuvre des connaissances devaient ensuite être validés lors de la quatrième étape.

La validation impliquait une évaluation par un expert des connaissances existantes dans les bases d'une part, des formalismes de représentation de ces connaissances, d'autre part. Il importait alors que l'expert puisse apprécier l'efficacité de l'outil, à l'aide d'un jeu de tests réalisés sur des exemples de vues graphiques industrielles réelles ou simulées comprenant des erreurs de conception commises intentionnellement. Dans le cas le plus favorable, l'étape conduisait à l'adoption des règles testées. Sinon, elle entraînait la correction ou l'affinage des règles par leur mise à jour dans les bases. Elle pouvait aussi déboucher sur l'énoncé de nouveaux concepts et règles ou, dans le cas extrême, conduire l'expert à aborder de nouveaux thèmes nous ramenant à la première étape.

Notons que cet intérêt porté au domaine de l'acquisition des connaissances depuis le début de nos recherches, nous a incité à publier certaines réflexions personnelles liant la conception de systèmes

d'assistance, la considération du temps et l'acquisition de connaissances, et proposant des voies de recherche dans ce cadre. Cette réflexion, appuyée par une doctorante de l'équipe de Bernard HOURIEZ, a fait l'objet de deux articles (KOLSKI et DURIBREUX, 1992 ; 1993). Suite à cette expérience acquise lors du projet SYNOP, le travail de recensement, de formalisation et d'intégration de règles dans les bases d'ERGO-CONCEPTOR et du M.D.I. a été facilité (KOLSKI, 1992a). De plus, les connaissances acquises ont été exploitées lors de trois projets décrits ci-après.

### ***2.2.1.2. Contribution à une expérience de conception, réalisation et évaluation d'une interface homme-machine par leurs utilisateurs***

Notre première contribution à la mise à disposition de développeurs de connaissances livresques et didactiques date de 1991 dans le cadre d'une collaboration avec GAZ de FRANCE. Avant de décrire le travail réalisé, il est utile de préciser le contexte original dans lequel il s'est situé.

Dans le but d'aider les conducteurs du réseau de gaz dans leurs prises de décision en cas d'incident ou de programmation de travaux, un système d'aide a été intégré dans la salle de télé-exploitation du réseau de gaz de l'agglomération urbaine de Lille (LOUBRADOU et al., 1988). Il utilise deux écrans graphiques : sur le premier, le conducteur modifie certains paramètres du réseau lors de sessions de simulation ; il peut comparer l'état simulé du réseau avec l'état courant disponible sur le second écran. Les conducteurs ont été impliqués dans les phases de conception et de réalisation du synoptique, ainsi que dans celle d'évaluation du système. Ce travail s'ajoutait à leurs tâches principales de conduite du réseau. Pour cela, ils disposaient (i) d'un éditeur de synoptiques, (ii) d'un SGBD permettant la description de chacun des éléments constituant le réseau de gaz. Les trois conducteurs, après avoir subi une formation spécifique devaient schématiser eux-mêmes le réseau de gaz de l'agglomération urbaine de Lille en se partageant le territoire, après avoir défini des symboles proches de ceux utilisés habituellement sur les plans papier. Une douzaine de vues ont été créées.

Dans un but d'évaluation et d'amélioration des vues réalisées, le responsable du projet a fait appel au LAIH, à un stade déjà avancé de la conception et de la réalisation des vues. On m'a confié la mission d'effectuer cette intervention. Ainsi, les conducteurs, après une formation de plusieurs journées sur la conception et l'évaluation, ont été amenés à étudier et critiquer l'interface qu'ils avaient eux-mêmes réalisée. Mon intervention s'est appuyée sur des méthodes de conception et d'évaluation développées au LAIH, ainsi que sur des connaissances recensées lors des projets SYNOP et ERGO-CONCEPTOR. Selon une approche *théorique*, ils ont d'abord été sensibilisés à l'ergonomie des interfaces homme-machine, en partant de concepts sur les utilisateurs d'interface pour aboutir progressivement à des recommandations concernant celle-ci. Cette sensibilisation a fait l'objet de nombreuses phases de discussion par rapport à la problématique de la conduite de gaz.

Une approche *pratique* a ensuite débouché sur une évaluation des vues réalisées en reprenant globalement puis point par point les recommandations étudiées. Cette évaluation a conduit à des propositions de réaménagement issues de l'ensemble des conducteurs. Suite à ces propositions, des essais ont été réalisés directement sur station. Des phases de discussions ont émergé de nombreuses propositions concernant l'amélioration de l'interface, l'objectif étant que celles-ci proviennent avant tout des conducteurs eux-mêmes. En raison du nombre peu élevé de conducteurs, il a été possible lors du projet de les rassembler lors des horaires normaux de travail. Une condition s'imposait : la formation devait être effectuée dans la salle de conduite et s'interrompre immédiatement en cas de problème. Cette contrainte s'est rapidement transformée en avantage pour la formation, dans la mesure où il était possible de discuter à chaud des problèmes ergonomiques et techniques rencontrés.

Cette assistance à la conception et à l'évaluation de l'imagerie s'est étalée d'avril à juin 1991. Cette expérience a été vraiment enrichissante, dans la mesure où il a été possible de côtoyer en direct les conducteurs lors de phases de résolution de problème, de confronter un ensemble de méthodes de travail et

de points de vue concernant la conception et l'évaluation d'interfaces, d'établir un dialogue avec les conducteurs qui changeaient en permanence de rôle, passant de celui d'utilisateur du système à celui de développeur, de réalisateur ou d'évaluateur de celui-ci. Une telle situation est fort peu courante et n'avait quasiment jamais été mentionnée à notre connaissance dans la littérature. Ce travail a fait l'objet d'un article de KOLSKI et WAROUX (1992). Un rapport prenant la forme d'un support d'intervention pédagogique a été rédigé pour l'approche théorique de l'intervention (KOLSKI, 1991).

### ***2.2.1.3. Analyse de besoins et recommandations pour le développement de systèmes de télémaintenance***

Dans les années 80 est apparue le principe de télémaintenance consistant en une maintenance à distance des installations, où un expert de télémaintenance utilise un système informatique à partir d'un site différent du site de production, figure 2.16. Ce système le plus souvent hautement interactif est composé d'un logiciel de dialogue et d'une console.



Figure 2.16 : Principe de base de la télémaintenance

Le L.A.I.H. a été sollicité par la société B+ DEVELOPMENT dans le cadre d'un projet du Ministère de la Recherche et de la Technologie pour rédiger un guide de conception de systèmes informatiques pour la télémaintenance d'installations automatisées, focalisé sur la communication homme-machine (KOLSKI et MILLOT, 1992b). Il consiste en un support méthodologique, basé sur des outils, techniques, modèles et recommandations pouvant s'appliquer à la majorité des applications actuelles.

Dans le but d'illustrer le guide d'exemples concrets issus d'applications réelles, nous avons procédé au préalable à des analyses de besoins d'utilisateurs de systèmes de télémaintenance pour trois applications différentes. La première analyse a été menée autour d'un système installé dans un centre de traitement des eaux usagées en Allemagne. La seconde a été centrée sur le système à base de connaissance DEEPEXPert d'aide à la supervision d'opérations de travaux sous-marins en plongée profonde. La troisième a concerné des machines de tri d'objets à haute cadence.

Ce projet a débuté en juin 1991 pour se terminer en mai 1992. Il a concerné un domaine pointu où les études sont encore rares. Ce projet a été particulièrement intéressant et formateur puisqu'il était nécessaire de se baser sur l'analyse de situations réelles aux problématiques sécuritaires ou économiques sévères, pour proposer ensuite des solutions concrètes utilisables par des développeurs de tels systèmes. Ce travail a fait l'objet de deux articles (KOLSKI et MILLOT, 1992a ; 1993).

### ***2.2.1.4. Mise à disposition de développeurs d'imagerie de connaissances livresques et didactiques***

Des recherches ont été menées dans le cadre d'une collaboration avec SOLVAY S.A., groupe spécialisé dans plusieurs secteurs chimiques. Il existe plusieurs dizaines de sites de production de produits chimiques dans le monde et souvent plusieurs salles de supervision sur chaque site. Les besoins de

SOLVAY en matière de développement d'imageries sont donc énormes. A ce sujet, l'étude s'est inscrite dans une démarche visant à mettre à la disposition de développeurs des connaissances utilisables pour la conception et l'évaluation d'interfaces pour les salles de supervision du groupe.

La démarche de travail est résumée en figure 2.17. Nous avons d'abord effectué un recensement de connaissances relatives à l'ingénierie des interfaces. En parallèle, il a été possible de visiter des salles de supervision, d'effectuer des entretiens avec des utilisateurs et des développeurs, et d'analyser des dizaines de pages-écrans. Ces données ont été rassemblées sous forme de fiches dans un guide (KOLSKI, 1992b). Chaque fiche exprime une idée, un concept ergonomique, et propose des recommandations le plus souvent illustrées de vues graphiques SOLVAY provenant de situations réelles (bons et mauvais exemples).

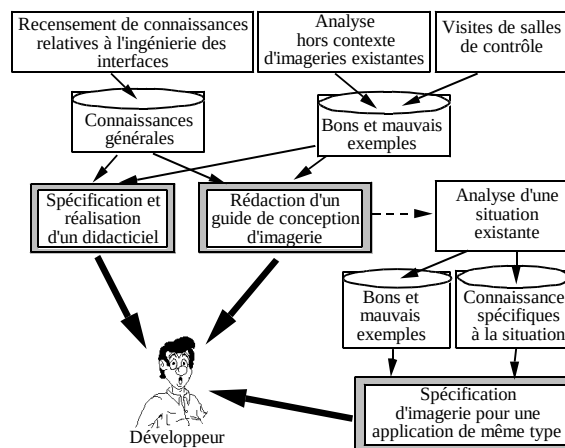


Figure 2.17. Démarche de travail

Par la suite, en nous inspirant du guide de conception rédigé, il nous a été demandé d'effectuer une analyse d'existant dans une salle de supervision d'une installation de fabrication d'eau oxygénée. Cette analyse avait pour but de fournir des spécifications d'interface à des développeurs de l'imagerie de supervision d'une nouvelle fabrication d'eau oxygénée. Il a fallu mettre au point une démarche d'analyse des besoins basée sur des entretiens, des questionnaires et des observations, en interaction avec des concepteurs, ainsi que quatre équipes d'utilisateurs aux stratégies souvent différentes (KOLSKI, RIERA, BERGER et MILLOT, 1993). Cette démarche est résumée en 2.2.3.

En parallèle, nous avons contribué à la spécification d'un didacticiel destiné à des développeurs d'imageries. Cette spécification porte principalement sur la démarche d'enseignement modélisée dans le didacticiel, et sur l'ensemble des connaissances décrites sous la forme de pages-écrans regroupées en un ensemble de thèmes relatifs à la conception d'imagerie. Ce didacticiel a été réalisé par SOLVAY sur PC à l'aide du logiciel TOOLBOOK. Ce didacticiel est original. A notre connaissance, il n'en existe pas dans le domaine de l'ingénierie des interfaces pour la supervision de systèmes complexes.

Ce projet a débuté fin 1991 pour se terminer fin 1993. Il a constitué une expérience très enrichissante dans la mesure où, malgré l'extrême richesse des concepts servant de base théorique au projet, sa finalité était très pragmatique et devait déboucher sur des connaissances pratiques, exprimées pédagogiquement afin d'être utilisables directement par les développeurs de SOLVAY.

#### 2.2.1.5. Conclusion sur le recensement de connaissances expertes

Il est important de centraliser et d'opérationnaliser au maximum l'expertise existante dans tous les domaines de l'ingénierie des interfaces, et ceci quelles que soient les applications visées. C'est pourquoi différents projets de mise à disposition de développeurs, de connaissances livresques et didactiques, ont naturellement découlé de nos recherches menées en ingénierie.

D'autres chercheurs travaillent dans ce domaine depuis plusieurs années, et des guides et des ouvrages voient progressivement le jour. Cependant, les documents actuels sont encore insuffisants, et appellent de nombreuses autres études et remises en question. Ils ne constituent qu'un pas dans ce domaine qui demande un énorme effort pédagogique de la part des spécialistes, pour effectuer un transfert efficace de compétence vers des milliers de développeurs d'interfaces homme-machine en attente de recommandations et de méthodes de travail concrètes.

### 2.2.2. Positionnement des méthodes d'évaluation "statique" automatique et de génération automatique de spécifications d'interface dans la méthode en U

Dans le chapitre précédent (cf. 1.4.3.2.1), les systèmes SYNOP et ERGO-CONCEPTOR ont été positionnés dans les *modèles formels de qualité de l'interface* contribuant à l'évaluation des interfaces homme-machine. Plus précisément, rappelons que SYNOP a été classé dans les *modèles dits de qualité de l'interface* s'intéressant à la *complexité perceptive des affichages*. ERGO-CONCEPTOR quant à lui a été classé dans les *modèles dits de qualité de l'interface* visant la *génération automatique d'affichage*.

Cependant, comme nous l'avons déjà précisé, l'évaluation résulte d'un état d'esprit se retrouvant dans toutes les étapes du développement. Ainsi, ces deux systèmes peuvent être situés en fait dans la méthode en U dans les étapes de spécification et de réalisation assistées de règles ergonomiques (figures 2.18 et 2.19). Dans ces étapes, on suppose connus les besoins informationnels des utilisateurs ainsi que les modes de coopération mis en place. Mais les méthodes axées sur l'utilisation de SYNOP et ERGO-CONCEPTOR sont fondamentalement différentes. En effet, SYNOP intervient seulement après l'édition graphique des images (figure 2.18).

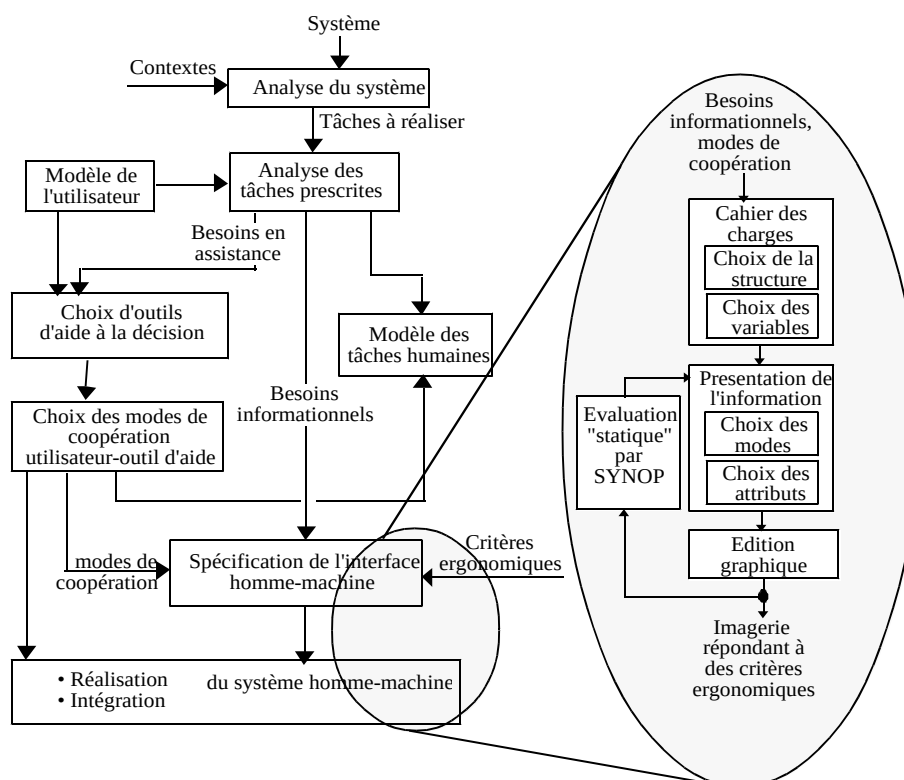


Figure 2.18 : Positionnement de la méthode d'évaluation "statique"

Au moment de l'intervention de SYNOP, la structure de l'imagerie, les variables, les modes de représentation et les attributs graphiques ont déjà été définies par le concepteur. L'intervention de SYNOP vise alors à évaluer la présentation de l'information selon les critères centralisés dans ses bases de connaissance. Cependant, les améliorations effectuées et les recommandations apportées ne concerneront pas la structure de l'imagerie, ni le choix des variables. Aucun mauvais choix à ce niveau ne sera corrigé. L'intérêt d'une approche d'évaluation "statique" de l'imagerie se limite donc à la manière dont l'information est présentée. Il est donc évident que d'autres méthodes devront être utilisées pour évaluer les aspects de l'interface non pris en compte. Ceci est prévu dans la phase ascendante d'évaluation prévue dans la méthode en U.

ERGO-CONCEPTOR par contre intervient avant l'édition graphique et vise à générer "automatiquement" des spécifications d'interface concernant la structure de l'imagerie, les variables, les modes de représentation et les attributs graphiques (figure 2.19). Les spécifications générées sont ensuite mises à la disposition du développeur. Ainsi, les vues générées sont sensées répondre à un ensemble de critères ergonomiques contenus dans les bases d'ERGO-CONCEPTOR. Par la suite, les vues sont évaluées dans la phase ascendante prévue dans la méthode en U.

Les méthodes à base de connaissances ergonomiques mises en place autour de SYNOP et ERGO-CONCEPTOR doivent encore être testées et validées sur de nombreux exemples industriels. Elles nous paraissent en tout cas représentatives de celles qui seront progressivement mises à la disposition des équipes de développement d'imageries industrielles.

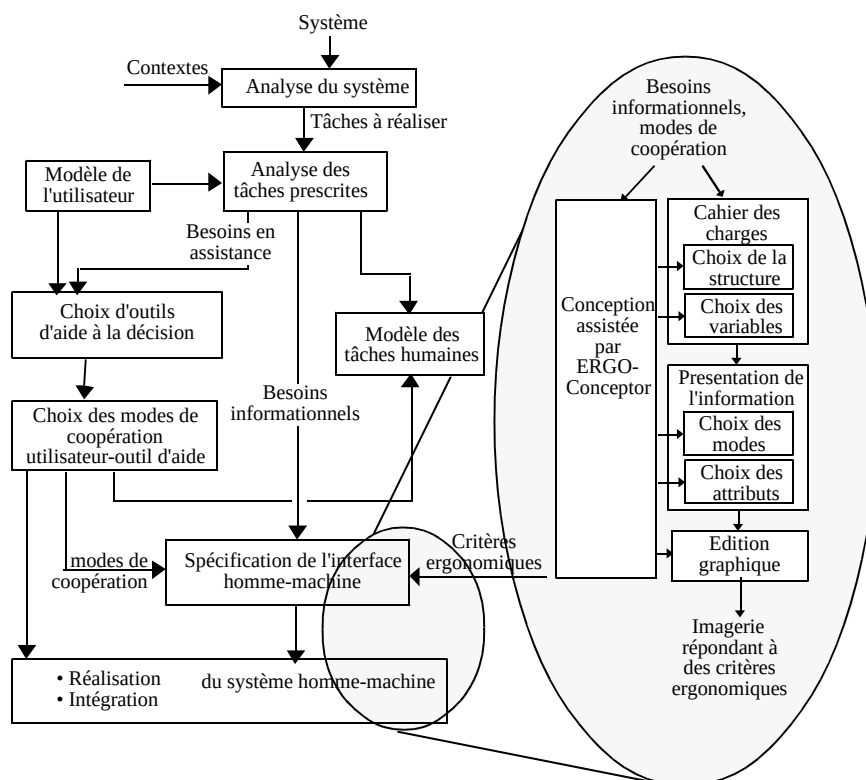


Figure 2.19 : Positionnement de la méthode de génération automatique de spécifications

### 2.2.3. Méthode rapide d'analyse de système existant pour la spécification d'imagerie de même type

Rappelons que l'une des tâches qui nous avait été confiées lors de la collaboration industrielle avec SOLVAY était une analyse d'existant menée dans une salle de supervision d'installations chimiques (Cf. 2.2.1.4). L'objectif à terme était de spécifier une nouvelle imagerie pour une autre usine (installations de même type), en tirant parti de l'expérience de l'existant.



En raison de la complexité du système homme-machine (plusieurs centaines de variables, environ deux cent vues graphiques, quatre équipes d'utilisateurs, plusieurs concepteurs et experts, critères de sécurité et de production) et du temps imparti (une vingtaine de jours-homme pour l'analyse), nous avons mis au point une démarche qualifiable de rapide. Celle-ci n'a aucunement la prétention d'analyser finement chacun des aspects du système homme-machine. Par contre, elle a permis de recenser énormément de données pertinentes. Elle commence par une sensibilisation par les experts au système (fabrication, imagerie, organisation du travail, etc), puis s'articule en deux étapes, figure 2.20 :

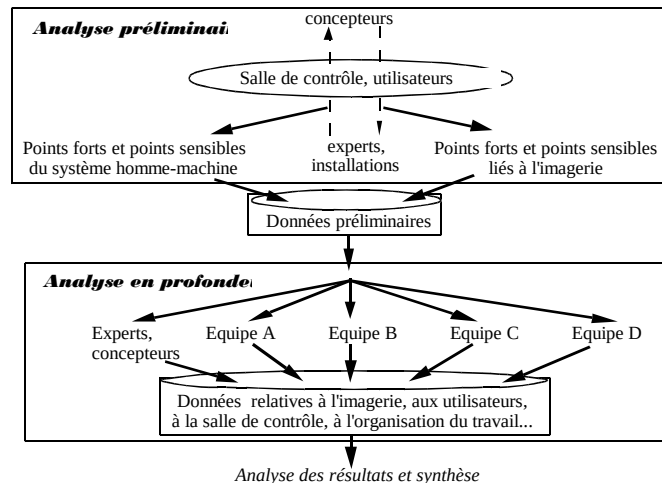


Figure 2.20. Démarche décomposée en deux analyses successives

- La première consiste en une analyse préliminaire permettant d'acquérir des notions de base sur les installations et de soulever des pistes de recherche susceptibles d'être intéressantes.
- La deuxième étape consiste en une analyse en profondeur. Au moyen d'un outil de travail de type questionnaire, elle vise à extraire des connaissances des personnes travaillant en lien plus ou moins direct avec la salle de supervision (utilisateurs, ingénieurs, concepteurs).

### 2.2.3.1. Analyse préliminaire

Afin de faciliter la recherche des points forts et des points sensibles liés à l'imagerie, la phase d'analyse préliminaire a séparé l'intervention en deux approches menées en parallèle : une première menée dans la salle de supervision et une seconde menée hors salle :

- Pour acquérir l'information nécessaire à l'évaluation, les utilisateurs et les ingénieurs de production ont d'abord été interrogés. Cette approche est intéressante car elle permet d'une part d'étudier le système avec les personnes les plus concernées et d'autre part d'avoir une vision générale des interactions homme-machine existantes. Cette approche est dénommée ascendante car elle part de la salle pour aller vers les concepteurs, c'est-à-dire des fonctions existantes pour aller vers les fonctions à assurer.
- En parallèle, les concepteurs ont été rencontrés, et l'imagerie a été étudiée statiquement. Cela permet de comprendre les choix d'imagerie et les problèmes rencontrés lors de la conception. On obtient ainsi l'avis des concepteurs, qui peut se révéler dans la pratique bien différent de celui des fabricants. De plus, l'étude en statique permet de mettre en évidence les caractéristiques de l'imagerie aussi bien en termes de points forts que de points sensibles. Cette approche est dénommée descendante car on part des concepteurs, et donc des fonctions à assurer pour aller vers la salle de supervision.

L'analyse préliminaire a fourni un premier ensemble de données brutes très riches. Par exemple, la structure de l'imagerie était bien perçue par les utilisateurs, mais certains choix allaient à l'encontre de règles de l'ergonomie des logiciels. La réalisation des images était rébarbative pour les concepteurs, rendant quasi-impossible le maquettage ou le prototypage. La salle de supervision était "humaine" et facilitait l'échange d'information sur la fabrication. D'ailleurs les utilisateurs n'étaient pas constamment stressés et avaient pu communiquer avec nous durant toute la durée de l'analyse. De plus, il semblait que l'organisation de travail était efficace et évitait une dégradation du modèle mental que les utilisateurs s'étaient forgé. Mais les alarmes posaient de nombreux problèmes. Enfin, les utilisateurs ne possédaient pas une connaissance exhaustive des possibilités de l'imagerie. Les points forts et points faibles mis en évidence ont ensuite été approfondis lors d'une analyse en profondeur.

#### **2.2.3.2. Analyse en profondeur**

Dans la démarche adoptée, il s'agissait de confirmer, d'infirmer ou de préciser les données recueillies lors de l'analyse préliminaire. Dans ce but, nous avons mis au point un outil de travail de type questionnaire pour aller plus en profondeur dans l'analyse. Il commence par une série de questions facilitant la prise de contact avec l'utilisateur interrogé. Puis, il se compose de cinq parties. Il commence par une série de questions concernant l'importance des variables et des vues. Dans la seconde partie, on trouve un ensemble de questions spécifiques à chaque mode de représentation de l'imagerie. Dans la troisième, on trouve des questions visant l'évaluation globale de l'imagerie, selon plusieurs critères de l'ergonomie des logiciels. Dans la quatrième partie ont été recensées des questions relatives aux aides fournies aux superviseurs pour utiliser l'imagerie. Enfin, la cinquième permet d'envisager avec les utilisateurs des améliorations de l'imagerie.

Le questionnaire n'a pas été utilisé au sens restrictif du terme, c'est-à-dire qu'il n'a pas été fourni aux superviseurs pour être rempli par ceux-ci. C'est pourquoi le terme questionnaire ne le caractérise pas exactement. Son usage par les analystes a plutôt été celui d'un outil d'entretien pour *trouver et explorer de nouvelles pistes* conduisant à des points forts et des points sensibles de l'imagerie. Il a d'abord été testé avec un ingénieur de la fabrication. Puis, les questions ont été posées aux superviseurs de chacune des quatre équipes durant leur travail. Le fait de mener les entretiens dans la salle a permis d'accélérer le processus de recherche de pistes, dans la mesure où chaque remarque pouvait immédiatement être étayée d'exemples concrets. Les entretiens ont souvent été entrecoupés de manipulations de l'interface, sources de commentaires supplémentaires.

L'analyse en profondeur a débouché sur énormément de points forts et de points faibles qui ont ensuite été validés par les experts, les concepteurs et les utilisateurs, et rassemblés dans (KOLSKI, RIERA, BERGER et MILLOT, 1993). Au vu des résultats obtenus, la démarche de travail mise au point serait maintenant réutilisable pour d'autres analyses menées dans des conditions similaires.

#### **2.2.4. Méthode de création des bases du M.D.I.**

Dans la partie 2.1.4. nous avons décrit une approche d'interface "intelligente" appelée Module Décisionnel d'Imagerie (M.D.I.). Les règles manipulées par le moteur d'inférence du M.D.I. sont créées, compilées et maintenues grâce à une méthode mise au point au laboratoire se composant de quatre étapes, figure 2.21. La première conduit à un recensement de toutes les valeurs possibles liées à des critères de décision concernant la gestion de l'affichage par le M.D.I., par exemple : les degrés de gravité, les situations de fonctionnement, les besoins en outil d'aide liées aux tâches à accomplir par l'utilisateur, etc. Ces informations définissent un ensemble de descripteurs caractérisant le système homme-machine. Un descripteur est un couple (attribut, valeur). Par exemple, à l'attribut "Situation de Fonctionnement", peuvent être associées plusieurs valeurs, telles que : normale, anormale et critique.

A partir de la base de faits possibles, nous avons mis en relation les critères de décision recensés avec les décisions possibles du M.D.I. Ces décisions concernent le "QUOI", le "QUAND" et le "COMMENT". Une base d'exemples est générée lors de cette étape. La description des exemples est effectuée à l'aide des attributs identifiés comme étant représentatifs du problème étudié. Chaque attribut possède un nombre fini de valeurs mutuellement exclusives.

Sur la base d'exemples, nous utilisons des techniques issues de l'apprentissage symbolique automatique (GANASCIA, 1990 ; MICHALSKI et KODRATOFF, 1990) pour générer des arbres de décision optimisés à l'aide du système 1stClass. Celui-ci se base sur l'algorithme ID3 (Interactive Dichotomizer 3), qui engendre à partir d'une collection d'exemples, appelée ensemble d'apprentissage, des procédures de classification exprimées sous la forme d'arbres de décisions (QUINLAN, 1983). Des arbres de décision sont visibles dans la thèse de TENDJAOUI. La dernière étape a pour but la compilation de ces arbres sous un format exploitable par le M.D.I. Ceux-ci sont transformés en règles de production utilisées par le moteur d'inférence.

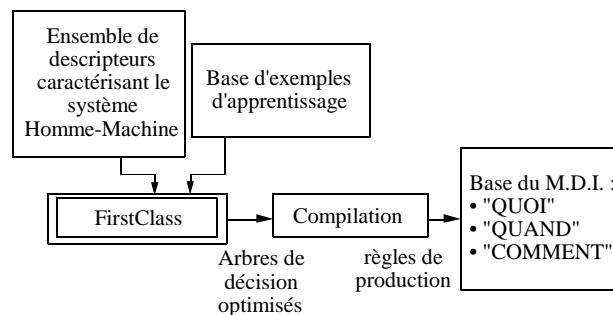


Figure 2.21 : Méthode de construction des règles

Même si cette méthode doit encore être améliorée par la mise en place d'outils plus intégrés et plus ergonomiques d'exploitation par le développeur des bases, elle présente plusieurs avantages. D'abord, dans la mesure où le M.D.I. doit composer et enchaîner les vues en fonction d'un ensemble de critères, il doit définir en fait les décisions à prendre pour répondre aux questions "QUOI" présenter, "QUAND" et "COMMENT" présenter l'information. C'est pourquoi l'utilisation d'arbres de décision nous a paru adaptée. De plus, l'utilisation d'un algorithme de type ID3 accélère la phase de création des arbres à partir d'exemples entre lesquels les liens ne sont pas toujours évidents. Enfin, en cas d'extension des descripteurs faisant office de critères de décision (nouveaux attributs et/ou nouvelles valeurs), l'algorithme se charge de construire les arbres à partir des exemples mis à jour, arbres qu'il s'agit ensuite de valider.

### 2.2.5. Méthode de développement d'imagerie axée sur le prototypage

Dans le cadre du projet ATLAS (cf. 2.1.3), il a fallu proposer une démarche complète supportée par l'atelier pour faciliter la spécification, la réalisation et la validation d'imageries (figure 2.22). Celle-ci s'inspire de la méthode en U (cf. 1.1.2) et comprend deux phases. La première, qualifiée de "descendante" consiste à partir des connaissances du système à superviser et des besoins informationnels des utilisateurs pour aboutir au dossier de définition du logiciel de supervision. La seconde phase, qualifiée "d'ascendante", vise à produire l'application qui sera implantée sur le site. On peut distinguer dans cette approche plusieurs étapes, résumées ci-dessous :

- *Description des installations*: cette étape conduit à décrire et à définir les installations selon des aspects fonctionnels et structurels, et ceci selon différents niveaux d'abstraction. Cette étude doit permettre de mieux appréhender d'une part les installations à superviser, mais aussi d'extraire la connaissance des experts ainsi que celle des futurs utilisateurs de l'imagerie. Il s'agit de fournir au spécifieur des fonctionnalités lui permettant de décrire facilement l'installation, à partir d'une

bibliothèque de composants types. Il est important que cette description serve par la suite de base à la définition et la réalisation des images.

- *Définition du prototype* : cette étape consiste à spécifier les différentes images composant le synoptique. Celle-ci requiert, d'une part, la connaissance des installations issue de l'étape précédente, et d'autre part les besoins informationnels des utilisateurs.
- *Réalisation du prototype* : lors de cette étape, le spécifieur doit disposer d'un outil lui facilitant la réalisation d'un prototype. Pour contribuer à une véritable démarche itérative, il doit être facile d'utilisation. De plus, la simulation du prototype doit être la plus réaliste possible, ceci dans le but de respecter la sémantique de l'application. C'est pourquoi la notion de *spécification graphique* (cf. 1.4.1.1) a été introduite dans cette étape, consistant à disposer sur l'écran les différents constituants de l'interface, à l'aide de la souris. Avant de passer à l'étape d'évaluation, on doit procéder à la génération du code exécutable de l'application de supervision à partir des données de spécification. Cette génération doit dans la mesure du possible être transparente pour le spécifieur et par conséquent automatique.
- *Evaluation ergonomique du prototype* : cette étape a pour objectif de tester les images réalisées lors d'un ensemble de situations simulées. Cette évaluation peut conduire à une modification de la définition ou de la conception du prototype.

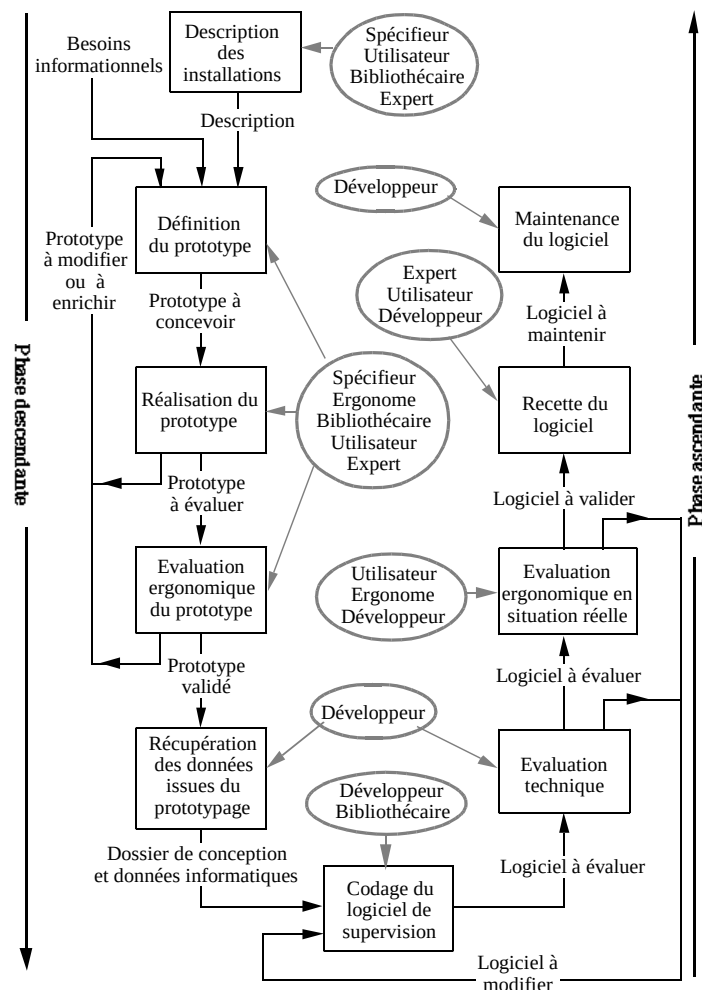


Figure 2.22 : Démarche de développement des images

- *Récupération des données issues du prototypage* : une motivation supplémentaire du développeur à utiliser la démarche réside dans la possibilité de récupérer au maximum le travail issu du prototypage. C'est pourquoi il faut veiller à engendrer une documentation du prototype (images, description de la sémantique de l'application...) et à interfacer d'un point de vue logiciel l'outil de prototypage avec ceux utilisés lors des étapes suivantes.
- *Codage du logiciel* : cette étape consiste à développer dans sa version finale l'application de supervision et à l'implanter sur le site. Dans ce but, le développeur reproduit d'abord les images dans le cas où il ne dispose que de documents papiers ou récupère les fichiers informatiques correspondant. Puis il modifie le comportement logique des constituants de l'image dans la mesure où une bibliothèque de composants simplifiés est utilisée. De plus le développeur doit préciser les moyens d'accès aux données provenant des installations. Une génération automatique maximale de code est préconisée dans cette étape.
- *Evaluation technique* : cette validation technique, menée par le développeur, consiste à tester les différentes fonctions informatiques, de manière à identifier et corriger les erreurs logicielles, et cela dans le but d'obtenir un produit techniquement fiable.
- *Evaluation ergonomique en situation réelle* : il est indispensable, une fois le logiciel opérationnel et implanté sur le site, d'effectuer une nouvelle évaluation. En effet, lors du prototypage, elle a été effectuée en situation simulée essentiellement.
- *Recette et validation du logiciel* : cette étape classique du génie logiciel a pour but de vérifier l'adéquation du logiciel aux besoins du client et donnera lieu à la mise en œuvre de scénarios de recette. Dans la mesure où la démarche proposée incite les différents acteurs à dialoguer, la recette doit se trouver inévitablement minimisée. En effet, dès les premières étapes, le spécifieur soumet pour évaluation les spécifications aux utilisateurs ainsi qu'aux experts, et de ce fait une détection précoce des problèmes est rendue possible.
- *Maintenance* : cette étape classique en génie logiciel consiste à maintenir le logiciel réalisé.

La démarche reposant sur le prototypage, elle permet d'évaluer les choix de conception dès les premières étapes de développement des applications de supervision selon une démarche itérative. Elle vise aussi à instaurer un dialogue entre les différents intervenants du processus de développement, ceux-ci ayant des compétences et des points de vue différents. Par conséquent, elle permet de confronter les opinions et facilite le transfert de connaissance d'un domaine à l'autre. Néanmoins une telle approche ne peut être applicable que si l'on dispose d'un outil informatique puissant et facile d'utilisation pour spécifier et réaliser le prototype. C'est pourquoi notre proposition a résidé dans une spécification graphique découlant de la description des installations (Cf. 2.1.3).

## 2.2.6. Conclusion sur les méthodes mises en place

La mise en place de méthodes autour de nos différentes contributions au domaine de recherche a pris une large part dans nos travaux. Sans souci d'exhaustivité, cette partie a donné un aperçu des méthodes très différentes auxquelles nous avons été confrontés.

Qu'elles concernent l'évaluation, la conception, l'analyse d'existant, la création de bases de connaissances ou le développement complet d'une imagerie, elles partent d'un souci constant d'opérationnalisation des approches visées. Malgré d'énormes difficultés liées aux connaissances et concepts à considérer, leur mise en place fournit de premiers résultats prometteurs, qu'il s'agira de confirmer lors des recherches à venir.

## CONCLUSION

Ce chapitre a décrit la majeure partie des travaux effectués personnellement ou sous ma direction sur le thème de l'ingénierie de conception et d'évaluation des interfaces homme-machine dans les systèmes industriels complexes. C'est selon trois points de vue que ces travaux ont été expliqués : (i) selon celui des systèmes réalisés, (ii) celui des connaissances mises en œuvre et enfin (iii) celui des méthodes mises en place. Ils reflètent l'état d'esprit qui a été le notre ces dernières années dans cette thématique.

Concernant les systèmes réalisés, nous sommes parvenus à des approches originales validées sur des applications industrielles réelles ou simulées. Trois d'entre elles (SYNOP, ERGO-CONCEPTOR et le M.D.I.) se basent sur des connaissances dans le domaine de l'ingénierie des interfaces homme-machine. Ces connaissances ont été intégrées dans les bases, selon des visées complémentaires : l'évaluation d'interfaces pour SYNOP, la conception pour ERGO-CONCEPTOR et la gestion dite "intelligente" pour le M.D.I. Au vu des résultats prometteurs obtenus, on peut s'attendre à terme à une opérationnalité de tels systèmes sur le marché de l'ingénierie des interfaces homme-machine. Ce n'est pas un hasard d'ailleurs si un nombre sans cesse croissant de chercheurs s'intéresse à ce domaine liant aspects graphiques et connaissances expertes.

Le quatrième système réalisé, consistant en un outil de prototypage intégré dans l'atelier ATLAS, n'explicite pas de connaissances au contraire des trois systèmes précédents. Par contre, il vise à opérationnaliser la démarche complète de développement d'imagerie en facilitant le dialogue entre les intervenants du projet par l'intermédiaire d'un outil commun : l'atelier. Cette démarche rejoint un courant de pensée actuel visant à faciliter la conception participative.<sup>16</sup>

Le recensement et l'exploitation de connaissances n'a pas servi que pour remplir les bases des systèmes développés. En effet, certaines d'entre elles ont été reformulées et restructurées pour être intégrées dans des guides destinés à des développeurs d'imageries, lors de collaborations industrielles avec GAZ de FRANCE, B+ Development et SOLVAY. Ces travaux constituent une avancée dans le domaine de l'ingénierie des connaissances. Il ne peuvent qu'être approfondis, l'apparition de nouveaux moyens d'interaction homme-machine (réalité virtuelle, interfaces multimodales...) entraînant systématiquement de nouveaux besoins en recommandations et méthodes de travail pour les concepteurs et évaluateurs.

En ingénierie des interfaces homme-machine, de nombreux aspects techniques et humains doivent être simultanément pris en compte. C'est pourquoi il a été nécessaire de situer chacune des approches des systèmes réalisés dans une méthode de conception, d'évaluation ou encore d'analyse. Chacune d'entre elle contribue à sa manière également au domaine de recherche. Mais, il reste encore beaucoup de travail à effectuer pour les évaluer dans des contextes différents et les améliorer progressivement.

Le chapitre suivant présente une synthèse sur le domaine, décomposée en plusieurs axes prenant la forme de perspectives pour une équipe de recherche.

---

<sup>16</sup> Voir à ce sujet le numéro spécial de la revue *Communications of the ACM* consacré à la conception participative (volume 36, numéro 4, juin 1993).

## Bibliographie

### Bibliographie de l'équipe

#### Evaluation "automatique" d'IHM

- BINOT C., KOLSKI C., ROGER D., WILLAEYS D. (1987).** *Graphic information evaluation and expert system in ergonomic conception of control views.* IECON'87 thirteenth annual IEEE Conference, Massachussets, 1-6 nov.
- KOLSKI C. (1989).** *Contribution à l'ergonomie de conception des interfaces graphiques homme-machine dans les procédés industriels : application au système expert SYNOP.* Thèse de Doctorat, 30 janvier.
- KOLSKI C., BINOT C., MILLOT P., ROGER D. (1988).** *Use of ergonomic concepts by an expert system for man machine graphic interface optimization.* Dans "Software for manufacturing", D. Kochan and G. Olling (Eds.), North-Holland, pp. 559-567.
- KOLSKI C., GRZESIAK F., MILLOT P. (1988).** *Outil de conception et d'évaluation de la page-écran utilisant la technologie des systèmes experts.* Congrès : Systèmes Experts et Télématique, Paris, 28-29 Janvier, EC2.
- KOLSKI C., MILLOT P. (1989).** *Démarches ergonomiques d'évaluation et de conception d'interfaces graphiques homme-machine à l'aide de techniques d'intelligence artificielle : évolutions vis-à-vis de l'ergonomie.* XXVème Congrès de la SELF "Evolutions technologiques et Ergonomie", Lyon, 4-6 Octobre.
- KOLSKI C., MILLOT P. (1991).** *A rule-based approach to the ergonomic "static" evaluation of man-machine graphic interface in industrial processes.* International Journal of Man-Machine Studies, 35, pp. 657-674.
- KOLSKI C., MOUSSA F., MILLOT P. (1988 et 89).** *Mise en oeuvre d'une base de connaissances ergonomiques exploitables par le système expert SYNOP pour l'aide à la conception d'images embarquées en automobile et évaluation.* Rapports établis dans le cadre de la convention RENAULT N°UV/14, LAIH, Juin 88 et Fév. 89.
- KOLSKI C., VAN DAELE A., MILLOT P., DE KEYSER V. (1988).** *Towards an intelligent editor of industrial control views, using rules for ergonomic design.* IFAC Workshop "Artificial intelligence in real-time control", Clyne Castle, Swansea, Great Britain, 21-23 September.

#### Génération "automatique" de spécifications

- KOLSKI C., MOUSSA F. (1991).** *Une approche d'intégration de connaissances ergonomiques dans un atelier logiciel de création d'interfaces pour le contrôle de procédé.* Quatrièmes Journées Internationales : Le génie logiciel et ses applications, EC2, Toulouse, 9-13 décembre.
- KOLSKI C., MOUSSA F., MILLOT P. (1990).** *Formalisation de connaissances ergonomiques dans le domaine de l'ergonomie de conception d'interfaces graphiques Homme-Machine.* XXVI Congrès de la SELF "Méthodologie et outils d'intervention et de recherche en ergonomie", Montréal, 3-6 Octobre.
- MILLET M. (1993).** *Evaluation ergonomique et propositions d'amélioration du système ERGO-CONCEPTOR.* Mémoire de projet de fin d'études de DESS ICHM, université de Valenciennes, mars.
- MOUSSA F. (1992).** *Contribution à la conception ergonomique des interfaces de supervision dans les procédés industriels : Application au système ERGO-CONCEPTOR.* Thèse de Doctorat, soutenue le 15 juillet 1992 à l'université de Valenciennes.
- MOUSSA F., KOLSKI C. (1991).** *ERGO-CONCEPTOR : système à base de connaissances ergonomiques pour la conception d'interface de contrôle de procédé industriel.* Technologies Avancées, vol. 2 (2), 5-14, décembre.
- MOUSSA F., KOLSKI C. (1992).** *Vers une formalisation d'une démarche de conception de synoptiques industriels : application au système Ergo-Conceptor.* Colloque ERGO-IA "Ergonomie et Informatique Avancée", 7-9 Octobre, Biarritz.
- MOUSSA F., KOLSKI C., MILLOT P. (1990a).** *Artificial intelligence approach for the creation and the ergonomic design of man-machine interfaces in control room.* Ninth European Annual Conference on "Human decision making and manual control", Varese, Italy, September 10-12.
- MOUSSA F., KOLSKI C., MILLOT P. (1990b).** *Etude de l'intégration d'un outil expert interactif ergonomique dans la démarche globale de conception d'interface graphique homme-machine.* Colloque International "l'Ordinateur, l'Homme et l'Organisation II", Nivelles, Belgique, 9-11 Mai.
- MOUSSA F., KOLSKI C., MILLOT P. (1992).** *A formal methodology for ergonomic design of Man-Machine interfaces.* 5th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Man-Machine Systems, The Hague, The Netherlands, June 9-11.

#### Outil de prototypage

- POULAIN T. (1994).** *Contribution du génie logiciel pour la conception et l'évaluation d'applications de supervision.* Thèse de Doctorat, université de Valenciennes, février.

- POULAIN T., GERME J.P., KOLSKI C. (1993).** *Un atelier de génie logiciel pour la spécification, la réalisation et l'évaluation de synoptiques industriels embarqués.* Génie Logiciel et Systèmes Experts, n° 31, pp. 58-70, juin.
- POULAIN T., KOLSKI C. (1992).** *Monde réel et monde virtuel, la problématique du contrôle de procédé par un opérateur humain.* Conférence Informatique 92 "L'interface des mondes réels et virtuels", Montpellier, 23-27 Mars.
- POULAIN T., KOLSKI C., MILLOT P. (1991).** *Contribution of the prototyping techniques for the ergonomic design of process control synopsis.* Tenth European Annual Conference on "Human Decision Making and Manual Control", Liège, Belgium, 11-13 November.
- POULAIN T., VILAIN B., KOLSKI C., MILLOT P. (1991, 92 et 93).** *Atelier de création de synoptiques industriels.* Rapports de contrat dans le cadre d'une collaboration entre CSEE, 3IP et le LAIH faisant suite à un appel d'offres de l'ANVAR, décembre 1991, mai 1992, juin 1993.
- VILAIN B., POULAIN T., KOLSKI C.** *Un atelier logiciel de spécification, réalisation et évaluation de synoptiques industriels embarqués.* En cours d'édition dans la revue : Automatique-Productique-Informatique Industrielle (APII).

<b>IHM dites "intelligentes"</b>
----------------------------------

- GAMBIEZ F., KOLSKI C., TANG X. MILLOT P. (1991).** *Man-machine interface for PREDEX, an expert system shell for process control and supervision.* Dans "Decision support systems and qualitative reasoning", M.G. Singh and L. Travé-Massuyès (Eds.), North-Holland, pp. 341-346.
- GAMBIEZ F., TENDJAOUI M., KOLSKI C., MILLOT P. (1990).** *Etude et mise en oeuvre d'un module d'interface pour l'outil PREDEX de développement de système expert de filtrage et de gestion d'alarmes pour les procédés continus.* Rapport final, dans le cadre de la convention LAIH/S2O, juin.
- KOLSKI C., LE STRUGEON E., TENDJAOUI M. (1993).** *Implementation of AI techniques for "intelligent" interface development.* Engineering Application of Artificial Intelligence, vol. 6, n° 4, pp. 295-305.
- KOLSKI C., TENDJAOUI M., MILLOT P. (1990).** *An "intelligent" interface approach.* Dans "Ergonomics of hybrid automated systems II", W. Karwowski and M. Rahimi (Eds.), Elsevier, pp. 587-594.
- KOLSKI C., TENDJAOUI M., MILLOT P. (1992).** *A process method for the design of "intelligent" man-machine interfaces : case study : "the Decisional Module of Imagery".* Inter. Journal of Human Factors in Manufacturing, 2 (2), 155-175.
- LE STRUGEON E., TENDJAOUI M., KOLSKI C. (1992).** *Knowledge specification and representation for an "intelligent" interface devoted to process monitoring and supervision.* IFAC/IFIP/IMACS International Symposium on Artificial Intelligence in real-time control, Delft, The Netherlands, June 16-18.
- MANDIAU R., KOLSKI C., MILLOT P. (1991).** *Perspective of Human-Machine system modelling by logical approach : Contribution of technics issued from Distributed Problem Solving.* Dans "Designing for Everyone vol. 1", Y. Queinnec, F. Daniellou (Eds.), Taylor & Francis, pp. 436-438.
- MANDIAU R., KOLSKI C., MILLOT P., CHAIB-DRAA B. (1991).** *A new approach for the cooperation between human(s) and assistance system(s) : a system based on intentional states.* World Congress on Expert Systems, Orlando, Florida, Dec. 16-19.
- TENDJAOUI M. (1992).** *Contribution à la conception d'interface "intelligente" pour le contrôle de procédés industriels : application au Module Decisionnel d'Imagerie.* Thèse de Doctorat, soutenue le 20 novembre 1992 à l'université de Valenciennes.
- TENDJAOUI M., KOLSKI C., GAMBIEZ F., MILLOT P. (1991).** *Towards an "intelligent" interface between decision aid tool and operator in control rooms.* Dans "Designing for Everyone vol. 1", Y. Queinnec and F. Daniellou (Eds.), Taylor & Francis, pp. 436-438.
- TENDJAOUI M., KOLSKI C., MILLOT P. (1990).** *Interaction between real-time aid expert system, intelligent interface and human operator.* International Symposium Computational Intelligence 90 "Heterogeneous knowledge representation systems", September 24-28, Milano.
- TENDJAOUI M., KOLSKI C., MILLOT P. (1991a).** *An approach towards the design of intelligent man-machine interfaces used in process control.* International Journal of Industrial Ergonomics, 8, pp. 345-361.
- TENDJAOUI M., KOLSKI C., MILLOT P. (1991b).** *Knowledge based interface approach for real-time aid expert system.* IFAC/IMACS SAFEPROCESS'91 Symposium, 10-13 september, Baden-Baden.

<b>Recensement de connaissances expertes</b>
--

- KOLSKI C. (1991).** *Assistance graphique à l'opérateur : concepts de base et recommandations ergonomiques.* Support d'intervention dans le cadre de la collaboration Gaz de France/LAIH, septembre.



- KOLSKI C. (1992a).** *Formalization approaches of ergonomic knowledge for "intelligent" design, evaluation and management of man-machine interface in process control.* IFAC Symposium on Intelligent Components and Instruments for Control Applications, Malaga, Spain, May 20-22.
- KOLSKI C. (1992b).** *Conception ergonomique d'interface opérateur en salle de contrôle : lignes directrices.* Guide de conception rédigé dans le cadre de la convention L.A.I.H./SOLVAY, Janvier.
- KOLSKI C., DURIBREUX M. (1992).** *Towards cognitive engineering workbenches considering the time for the "intelligent" design of operator assistance tools.* Dans "Advances in Industrial Ergonomics and Safety IV", S. Kumar (Ed.), Taylor & Francis, pp. 51-58.
- KOLSKI C., DURIBREUX M. (1993).** *Towards explicit consideration of time in cognitive engineering workbenches for "intelligent" assistance system design.* International Journal of Industrial Ergonomics, 11, pp. 125-134.
- KOLSKI C., MILLOT P. (1992a).** *Decision aid criteria to integrate a telemaintenance tool into the maintenance man-machine system.* Dans "Advances in Industrial Ergonomics and Safety IV", S. Kumar (Ed.), Taylor & Francis, pp. 51-58.
- KOLSKI C., MILLOT P. (1992b).** *Télmaintenance de systèmes automatisés, guide ergonomique de conception.* Rapport final, convention L.A.I.H./B+ Development, mai.
- KOLSKI C., MILLOT P. (1993).** *Problems in telemaintenance and decision aid criteria for telemaintenance system design.* International Journal of Industrial Ergonomics, 11, pp. 99-106.
- KOLSKI C., RIERA B., BERGER T., MILLOT P. (1993).** *Analyse en salle de contrôle du procédé H2O2 à JEMEPPE, lignes directrices pour la conception d'imagerie pour un procédé de même type.* Rapport final établi dans le cadre de la convention LAIH/SOLVAY, septembre.
- KOLSKI C., WAROUX D. (1992).** *Une expérience de conception, de réalisation et d'évaluation par des conducteurs de leur propre synoptique de conduite de réseau gazier.* Le Travail Humain, tome 55, n°4, 371-388.
- Bibliographie complémentaire**
- BEBIN J. (1984).** *Génie Nucléaire.* Techniques de l'Ingénieur, Edition N°1063, Mai.
- GANASCIA J.G. (1990).** *L'âme-machine - les enjeux de l'intelligence artificielle.* Editions du seuil, Sciences ouvertes, Paris.
- GRZESIAK F. (1987).** *Représentation des connaissances et techniques d'inférence pour le maniement d'objets graphiques : application au système expert SYNOP.* Thèse de Docteur-Ingénieur, université de Valenciennes, Mars.
- HOURIEZ B. (1994).** *Acquisition de connaissances pour l'aide à la conduite et la supervision de procédés industriels.* Rapport d'habilitation à diriger des recherches, université de Valenciennes, février.
- LOUBRADOU P., RINGOT S., SAGER J.P., TRONC J.C. (1988).** *Un système d'aide à la conduite d'un réseau de distribution de gaz naturel.* Actes du congrès de l'Association Technique du Gaz, Paris, juin.
- MICHALSKI R.S., KODRATOFF Y. (1990).** *Research in Machine Learning, recent progress, classification of methods and future direction.* In Machine Learning : An artificial Intelligence Approach, vol. 3, Y. Kodratoff and R.S. Michalski (Eds.), Palo Alto, California.
- MINSKY M. (1975).** *A framework for representing knowledge.* In "The Psychology of Computer Vision", Editions P.H. Winston, Mc Graw Hill, New-York, pp. 211-280.
- PITRAT J. (1990).** *Métaconnaissance, futur de l'IA.* Editions Hermès, Paris.
- QUINLAN J.R. (1983).** *Learning efficient classification procedures and their application to chess end games.* In Machine Learning, an artificial Intelligence Approach, Vol. II, R.S. Michalski, J.G. Carbonnel and T.M. Mitchell (eds), Academic Press.
- SFERCA (1987).** *IMAGIN : documentation technique.* SFERCA, Villeneuve d'Ascq.
- TABORIN V., POULAIN T., MILLOT (1986, 1988, 1990).** *ALLIANCE : Système de gestion d'alarmes utilisant les techniques de l'Intelligence Artificielle. Spécifications de l'interface Homme-Système expert.* Rapports de contrat MRES-ADI, en collaboration avec le CEA, LAG, ITMI, IIRIAM, SHELL Recherche, EDF, SGN, juillet, novembre 1986, mai 1988, janvier 1990.

## Chapitre 3

### *Synthèse et perspectives de recherche autour du modèle $\nabla$ (Nabla)*

Dans le premier chapitre, au vu des lacunes actuelles des méthodes et modèles du génie logiciel, un cadre théorique et méthodologique a été défini. Celui-ci, prenant la forme d'une méthode dite en U, a d'abord permis de positionner un ensemble de concepts fondamentaux vis-à-vis de la conception et de l'évaluation de systèmes interactifs.

Puis, dans le second chapitre, c'est par rapport à ce cadre qu'ont été définies nos principales contributions au domaine. Rappelons que ces contributions ont été décrites selon trois points de vue : celui des systèmes réalisés selon des principes originaux, celui de la connaissance exploitée et celui des méthodes mises au point. Les recherches entreprises depuis 1985 ont débouché sur de premiers résultats prometteurs. Mais celles-ci ne constituent d'après nous qu'un point de départ nécessitant de nombreux approfondissements et remises en causes.

L'objectif de ce troisième chapitre est de tirer les leçons méthodologiques de nos différentes contributions en vue de déterminer la thématique de base d'une équipe de recherche. C'est dans cette optique et selon le même état d'esprit que quatre parties composent ce dernier chapitre.

La première partie vise à tirer l'essence des modèles de développement issus du génie logiciel et de la méthode en U dans le but de proposer un nouveau cadre théorique et méthodologique pour les années à venir. Ce cadre prend la forme d'un modèle appelé  $\nabla$  (lire "nabla").<sup>17</sup>

---

<sup>17</sup> Le caractère  $\nabla$  a été choisi pour son mimétisme graphique avec la démarche illustrée par le modèle.

Ce modèle intègre les étapes classiques du génie logiciel, mais il est destiné au développement de systèmes interactifs. Comme on l'a vu dans l'état de l'art, le développement des interfaces homme-machine nécessite un savoir faire et des moyens particuliers. C'est pourquoi les aspects liés aux interfaces ressortent clairement, au contraire des modèles cascade, en V et spirale, qui restent très généraux.

La seconde partie met en évidence l'idée suivante : le modèle  $\nabla$  offre un cadre dans lequel un retour d'expérience peut s'explicitier, et ceci selon le principe de réutilisabilité. Notre proposition consiste à faire apparaître deux aspects de mise à disposition de l'équipe de développement d'éléments réutilisables. Le premier niveau de réutilisabilité concerne la réalisation des interfaces homme-machine, alors que le second est en rapport avec d'éventuels modules d'aide.

Le modèle  $\nabla$  constitue essentiellement un cadre théorique et méthodologique. Il est donc hors de question de mettre tel quel à la disposition d'équipes de développement, même s'il fournit d'ores et déjà un ensemble de lignes directrices. C'est pourquoi la troisième partie discute de la mise en place de méthodes basées sur le modèle  $\nabla$ .

La quatrième partie s'intéresse à la nécessité de mettre en place des outils et ateliers logiciels, de développement de systèmes interactifs, supportant le modèle  $\nabla$ . Ceux-ci devront à l'avenir fournir l'ensemble des moyens nécessaires pour la couverture de toutes les étapes du cycle.

Enfin, la cinquième partie montrera que le modèle  $\nabla$  constitue une étape supplémentaire vers la mise en œuvre d'interfaces dites "intelligentes".

### 3.1. CYCLE DE VIE DE SYSTEME INTERACTIF ET NOUVEAU CADRE THEORIQUE ET METHODOLOGIQUE : LE MODELE $\nabla$

Nos perspectives de recherche s'articulent autour d'un modèle de développement de système interactif appelé modèle  $\nabla$ . La première version de ce modèle est visible en figure 3.1.

L'objectif du modèle est de situer les différentes étapes du génie logiciel nécessaires pour développer un système interactif, tout en différenciant l'interface proprement dite (partie gauche du modèle) des modules d'aide éventuellement accessibles à partir de ceux-ci (partie droite). Comme nous l'avons souligné dès l'introduction générale de ce mémoire, ces modules d'aide peuvent être purement algorithmiques, mais peuvent aussi consister en des systèmes à base de connaissances. Une des caractéristiques marquantes du modèle est de positionner des étapes - inexistantes dans les modèles classiques du génie logiciel - où les facteurs humains devront être considérés par l'équipe de développement.

La description du modèle est succinctement la suivante. La première phase est classique en génie logiciel et marque le début du projet en donnant une **orientation** au travail à réaliser (objectifs, organisation du projet, contraintes, etc).

Ensuite, le modèle accorde énormément d'importance à l'analyse du système homme-machine lors du projet, analyse portant sur le système complexe, les tâches humaines et les utilisateurs. Rappelons que dans le chapitre 1, nous avons énormément insisté sur celle-ci, au sujet de laquelle d'ailleurs les recherches actuelles sont particulièrement actives. Dans l'esprit de la méthode en U (Cf. 1.2.2), la modélisation doit s'orienter sur :

- un *modèle réel* correspondant au système homme-machine existant, concret, avec ses contraintes, points forts et points faibles. Comme nous l'avons expliqué dans le premier chapitre, en 1.2.1, figure 1.6, trois cas peuvent être considérés. Lorsqu'il s'agit de mettre à jour un système homme-

machine existant pour aboutir à un nouveau système homme-machine (figure 1.6, a), la modélisation est bien entendu effectuée à partir du système existant. Lorsqu'il s'agit de créer un nouveau système homme-machine à partir d'autres systèmes déjà existants (figure 1.6, b), la modélisation découle d'une synthèse des données issues de chaque analyse. Lorsqu'il n'existe pas de système homme-machine existant et que le système est entièrement à concevoir (figure 1.6, c), ce modèle doit être conçu.

- un *modèle de référence* qui correspond à celui d'un système homme-machine dit idéal, en considérant les points de vue et besoins des différents intervenants concernés par le système homme-machine visé. Ce modèle doit en particulier lister un ensemble de critères devant être respectés. Comme nous l'avons déjà souligné, ces critères sont de natures diverses (sécurité des hommes, des installations, de l'environnement, production, ergonomie du logiciel, économie d'énergie...).

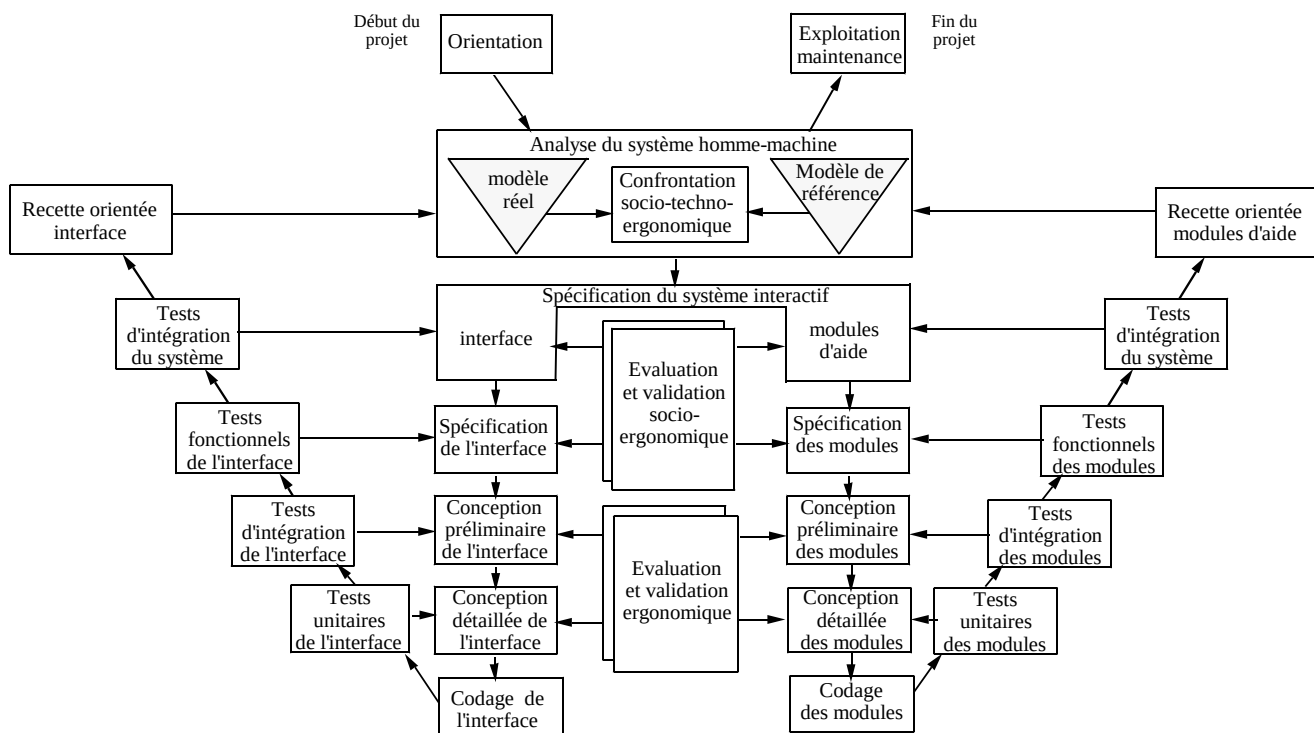


Figure 3.1. Modèle V de développement de système interactif

En confrontant progressivement les deux modèles au cours de l'analyse du système homme-machine, et en débouchant sur des compromis visant à satisfaire un maximum de critères, les données devraient être suffisamment pertinentes pour spécifier un système interactif adapté aux besoins informationnels des utilisateurs, ainsi qu'aux besoins en modes de coopération utilisateur-modules d'aide. Il s'agit ensuite de spécifier l'interface d'une part et les modules d'aide identifiés d'autre part. L'ensemble des spécifications devront être évaluées et validées d'un point de vue socio-ergonomique, afin de vérifier la pertinence de l'intégration des solutions nouvelles dans le système homme-machine visé, constitué la plupart du temps de plusieurs êtres humains, matériels et logiciels en inter-relation. Il s'agira de respecter l'aspect collectif du travail, aspect la plupart du temps négligé par les équipes de développement (ZOROLA-VILLARREAL et al., 1993).

Après avoir spécifié l'interface et les modules d'aide, et afin d'arriver à l'étape de codage, on procède de manière usuelle à la conception préliminaire et la conception détaillée, associées respectivement comme dans le modèle en V à des tests d'intégration, ainsi qu'à des tests unitaires.

En rapport avec la conception, il est important d'évaluer et de valider ergonomiquement les composants du système interactif. A ce sujet, notons l'importance de méthodes d'évaluation décrites dans le chapitre 1, dont certaines sont utilisables ici. Les tests de conception (Cf. 1.4.3.1.1) sont les plus connus, les modèles informels issus de l'approche analytique (Cf. 1.4.3.2.2) sont de plus en plus utilisés. Quant aux modèles formels issus de l'approche analytique de l'évaluation (Cf. 1.4.3.2.1), ils devraient à terme fournir des résultats spectaculaires et être intégrés progressivement dans les ateliers de génie logiciel. Notons que Martial GRISLIN devrait fin 1994 soutenir sa thèse sur l'apport des modèles formels dans le cycle de développement de systèmes interactifs. Comme l'indique le modèle  $\nabla$ , les enchaînements entre étapes sont propices à une démarche de prototypage, démarche dont l'importance a été précisée dans le premier chapitre.

Comme dans tout modèle existant, la phase de recette a été positionnée. Afin d'insister sur la problématique d'un système interactif, nous avons choisi de décomposer cette étape en la différenciant symboliquement en une recette orientée interface et une recette orientée modules d'aide. Ces deux étapes devraient se retrouver minimisées si le système interactif complet est conforme aux données issues de la modélisation du système homme-machine, et si chaque solution a été effectivement évaluée et validée.

Enfin, le cycle de développement se termine par une autre étape classique en génie logiciel, celle d'exploitation et de maintenance. Notons que cette étape pourrait reboucler sur l'étape d'orientation. Celle-ci pourrait en effet viser l'évaluation du système interactif en exploitation, dans un but d'amélioration. Ainsi, il s'agirait de procéder à une analyse du système homme-machine, débouchant sur de nouvelles spécifications au niveau de l'interface et/ou des modules d'aide. En cas de validation des spécifications, le reste du cycle pourrait être parcouru selon les principes décrits précédemment. En conclusion, ce modèle de développement peut être vu également comme un modèle d'évaluation et d'amélioration.

Le modèle  $\nabla$  s'inspire de nos différents travaux de recherche menés depuis 1985. Il devrait nous servir maintenant de nouveau cadre théorique et méthodologique, qu'il faudra affiner tout au long de ces prochaines années. C'est donc par rapport à celui-ci que plusieurs axes de recherche sont maintenant successivement précisés.

## **3.2. MODELE $\nabla$ ET REUTILISABILITE**

Dans ERGO-CONCEPTOR, c'est par l'intermédiaire de spécifications générées que le développeur se voit proposer un ensemble de modes de représentations (Cf. 2.1.2). Dans l'atelier ATLAS, le bibliothécaire doit gérer un ensemble de composants graphiques qu'il est ensuite possible d'appeler lors de la description des installations, puis pour la description des images (Cf. 2.1.3). Ces idées nous ont progressivement amené à nous interroger sur la notion de réutilisabilité de composants logiciels qu'il est intéressant de discuter par rapport au modèle  $\nabla$ . Mais d'abord, il convient de définir plus précisément le principe de réutilisabilité.

### **3.2.1. Le principe de réutilisabilité**

Appuyée par l'engouement provoqué par les travaux actuels concernant l'orientation objet, la réutilisabilité de composants logiciels, sous la forme de données et de traitements, constitue depuis la fin des années 80 un domaine de recherche à part entière (MEIJER et RAMES, 1989 ; IEEE SOFTWARE, 1990 ; HALL, 1992). Cette notion apparaît d'ailleurs pour des applications de supervision de systèmes industriels complexes. Par exemple, TRICOT et KATZ (1991) expliquent que dans un projet d'imagerie, sur 280 classes d'objets graphiques utilisés, 130 ont été jugés réutilisables, alors que les 150 autres étaient trop spécifiques pour être réutilisés. Ce nombre important de classes réutilisables (près de 50 %) nous paraît remarquable. Il constitue en tout cas une source potentielle de rentabilité.

Le principe de réutilisabilité est précisé sur la figure 3.2 par rapport à un cycle de développement logiciel (MOREL, 1994). Lorsqu'un composant logiciel est candidat à la réutilisabilité, il est (1) identifié et importé, (2) reconçu et qualifié selon le standard de réutilisabilité en vigueur. Puis, il est (3) classé et stocké dans une base logicielle qui (4) doit offrir des fonctionnalités de gestion. Lors d'un développement ultérieur, chaque composant (5) doit pouvoir être retrouvé dans la base, éventuellement évalué et comparé à d'autres composants également candidats à la réutilisation, puis adapté en fonction des besoins de l'application.

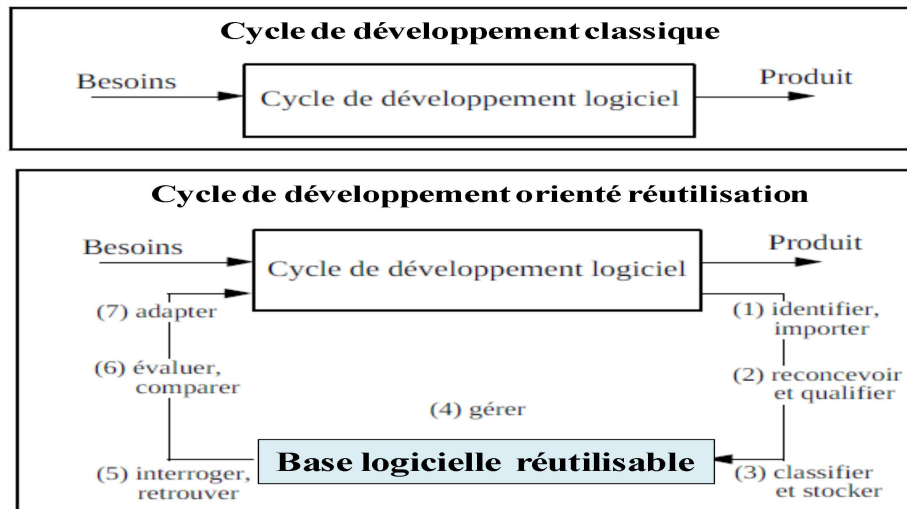


Figure 3.2. Cycle de développement et principe de réutilisabilité

A l'aide de la figure 3.2, on se rend compte aisément que le processus de réutilisabilité implique des activités spécifiques (gestion et utilisation d'une base) exigeant au départ un investissement important de la part de l'équipe de développement mais qui doit être rapidement rentabilisé, et sous-entend une nouvelle approche de travail dans les étapes du cycle.

Le principe de réutilisabilité constitue une perspective de recherche prometteuse permettant d'améliorer la capitalisation d'expérience lors du développement de systèmes interactifs. Si l'on se réfère au modèle  $\nabla$ , la réutilisabilité peut être naturellement envisagée selon d'une part les aspects liés à l'interface homme-machine, et d'autre part ceux liés aux modules d'aide éventuellement disponibles à partir de l'interface homme-machine.

### 3.2.2. Réutilisabilité au niveau des interfaces homme-machine

La réutilisabilité au niveau des interfaces homme-machine devrait pouvoir être envisagée selon trois niveaux, figure 3.3. Selon une démarche ascendante, le premier niveau vise, comme dans l'atelier ATLAS, la centralisation de composants de base réutilisables, par exemple : une courbe dont l'ordonnée est destinée à représenter une pression variant de  $V_i$  à  $V_f$  bars, ou le symbole d'une vanne associé à des règles de comportement graphique.

Au dessus de celui-ci peut être d'abord proposé un niveau où des "services" sont disponibles. Dans les systèmes industriels complexes, un service peut prendre la forme d'une fenêtre affichant la liste des messages d'alarmes, ou une fenêtre permettant de modifier une température de consigne.

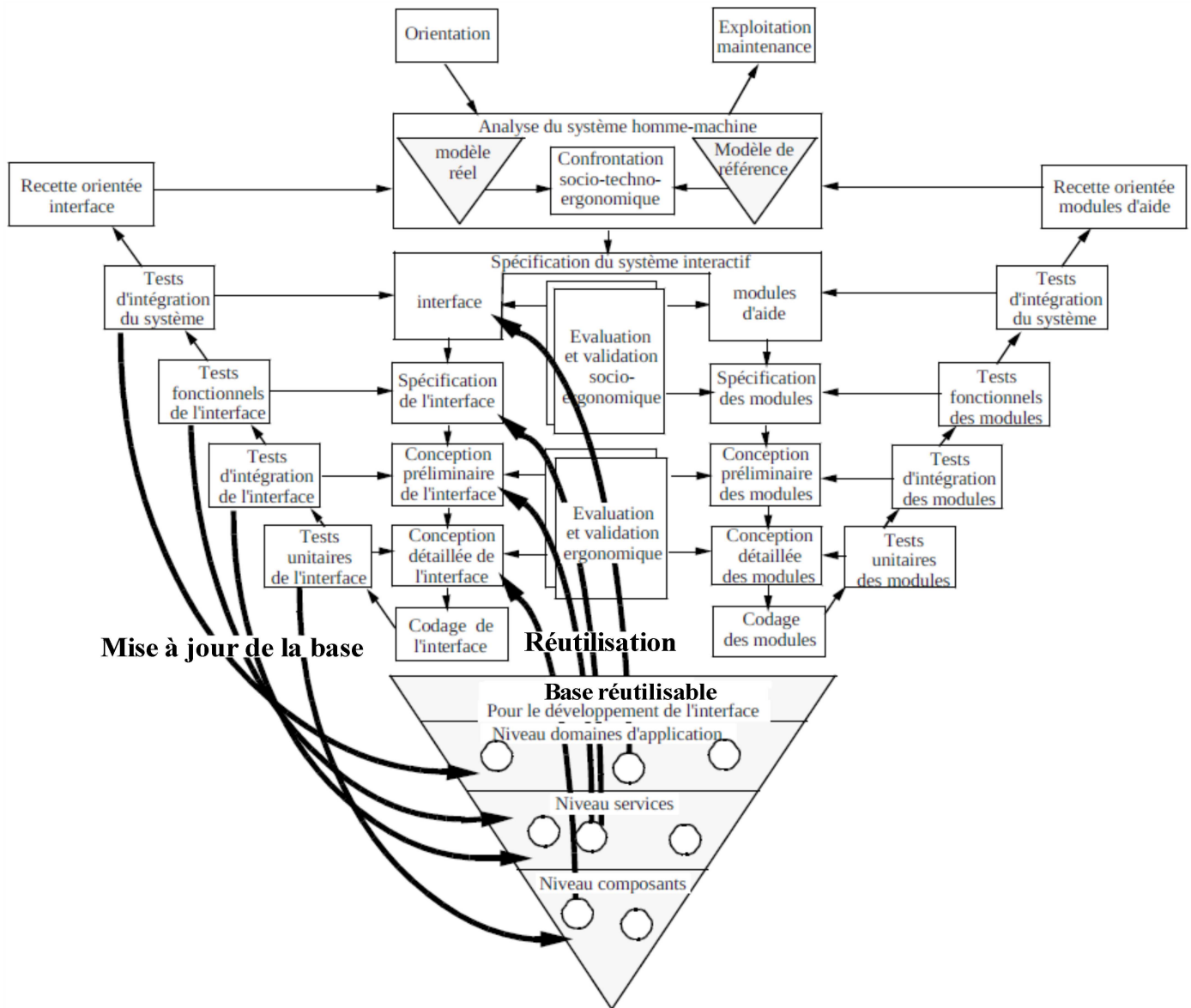


Figure 3.3 : Modèle V et principe de réutilisabilité pour le développement de l'interface

Au niveau le plus haut, appelé niveau des "domaines", il est possible de rendre disponibles différentes architectures d'imagerie ayant fait leur preuve dans des domaines différents. Par exemple, dans le domaine des systèmes industriels complexes peuvent être fournis des architectures relatives respectivement à la supervision d'un haut-fourneau, d'un laminoir, d'un atelier manufacturier, etc.

### 3.2.3. Réutilisabilité au niveau des modules d'aide

En intelligence artificielle, de nombreuses recherches sont menées actuellement autour de la modélisation de la connaissance dans un but de capitalisation d'expérience que ce soit sous forme d'outil d'EAO, de système à base de connaissance, voire même de document papier (CHANDRASEKARAN, 1987 ; VOGEL, 1988 ; ERMINE, 1993 ; WIELINGA et al., 1992 ; 1993 ; HOURIEZ, 1994). C'est pourquoi la réutilisabilité devrait pouvoir être prévue au niveau des modules d'aide également, particulièrement dans le cas où ceux-ci sont basés sur des techniques d'intelligence artificielle. Dans ce but, il nous paraît opportun de nous inspirer de la décomposition en vigueur dans les premières versions de la méthode KADS qui peut

être considérée comme la méthode de référence en matière de développement de système à base de connaissance, figure 3.4.<sup>18</sup>

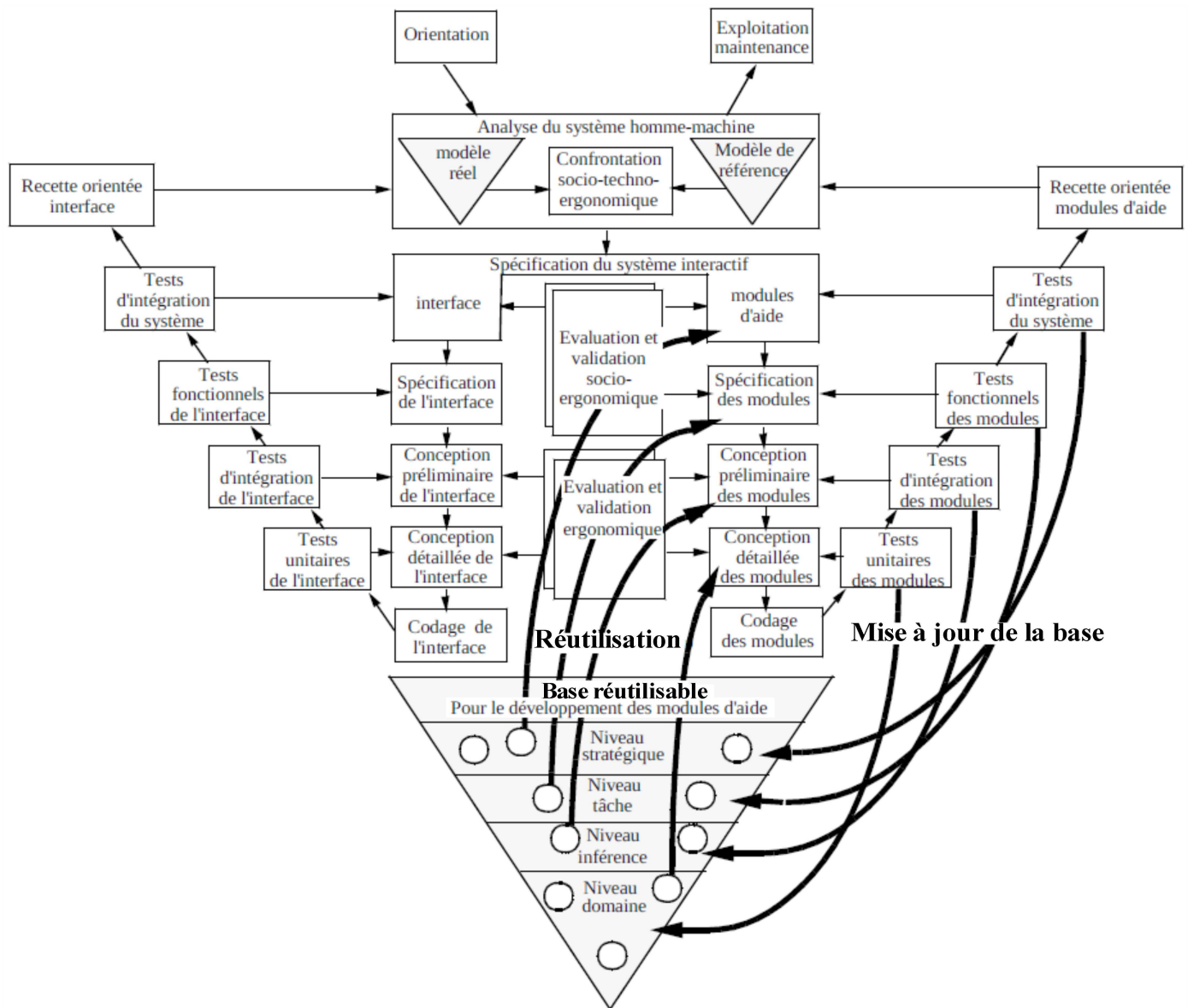


Figure 3.4 : Modèle V et principe de réutilisabilité pour le développement des modules d'aide

En respectant les niveaux de KADS, il apparaît un premier niveau appelé "domaine" qui correspond à la théorie du domaine et comprend des connaissances "statiques" sous forme de concepts, propriétés, relations... Le second niveau, appelé "inférence", concerne toutes les inférences qui peuvent être réalisées sur les entités du niveau précédent. Cette description est faite en termes de "rôles" (hypothèse, question, plan...) et de "fonctions" (instanciation, classification, assemblage, comparaison...) associés à la connaissance. Le troisième niveau, appelé "tâche", décrit comment effectuer et séquencer les inférences afin d'atteindre un but. Enfin, le niveau "stratégie" comprend des plans de résolution sous forme d'enchaînement de différentes tâches, de contrôle de réalisation de buts, de gestion des échecs d'une méthode de résolution, etc.

<sup>18</sup> La méthode KADS (Knowledge Acquisition and design Structuring) est issue des projets Esprit KADS I et KADS II (WIELINGA et al., 1992). Débouchant sur une version appelée Common-KADS (WIELINGA et al., 1993), elle est guidée par la construction de différents modèles de la connaissance. Elle est supportée par deux ateliers de génie cognitif dénommés KADS-Tools (Ilog, Cap Gemini Innovation) et Open-KADS (Bull).



Par exemple, pour des contextes de supervision de systèmes chimiques, on pourra trouver les éléments suivants dans les quatre niveaux. Au niveau "domaine" seront décrits des composants (vannes, cuves...) susceptibles d'appartenir à des lignes de production chimique avec leurs attributs (entrées, sorties...), les relations entre composants, etc. Au niveau "inférence" seront centralisés des "rôles" et des "fonctions" concernant les composants. Au niveau "tâche", on décrira par exemple comment parvenir à identifier un composant en défaut dans un sous-système. Au niveau "stratégie" seront recensés des plans d'assistance à l'utilisateur dans des contextes bien identifiés, ces plans étant basés sur les tâches décrites au niveau précédent.

### 3.2.4. Conclusion sur les perspectives en terme de réutilisabilité

Un travail de recherche de fond me paraît devoir être mené autour de la réutilisabilité de composants logiciels pour le développement de l'interface et des modules d'aide. Si l'on revient à la figure 3.2, on s'aperçoit que pour assurer les 7 services - allant de l'identification/importation à l'adaptation du composant logiciel - positionnés autour du cycle de développement, de nombreuses fonctions doivent être prévues, tout en restant les plus transparentes possibles pour l'équipe de développement.

Un travail important consiste à définir les jonctions entre les composants relatifs à l'interface et ceux concernant les modules d'aide, figure 3.5. En effet, le but est d'arriver à une base unique la plus cohérente et standard possible vis-à-vis des besoins de l'équipe de développement. Ce travail nécessite un rapprochement thématique avec des équipes travaillant en ingénierie de la connaissance et utilisant déjà des ateliers de génie cognitif.<sup>19</sup>

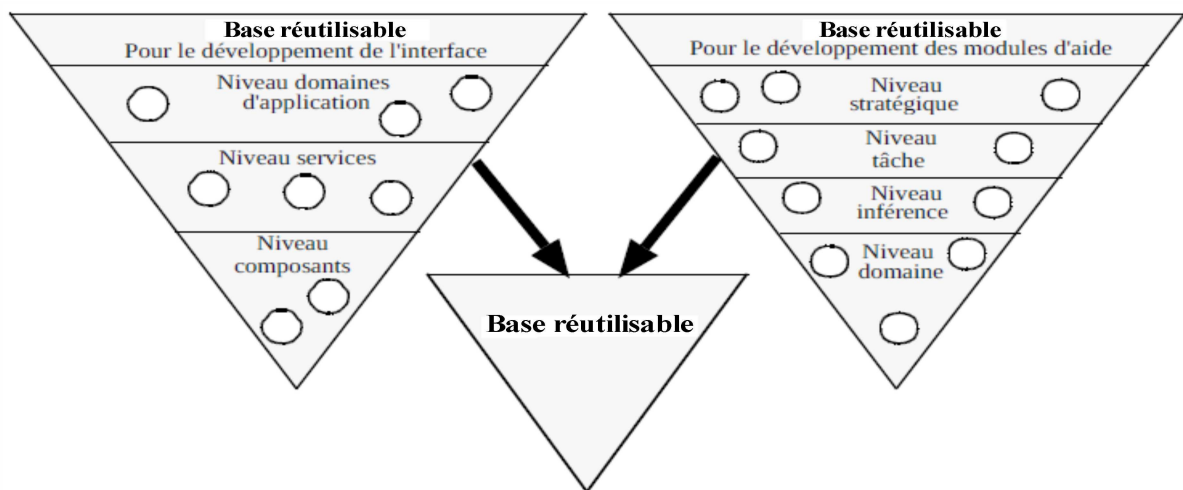


Figure 3.5 : Mise en place d'une base réutilisable unique

### 3.3. VERS DES METHODES BASEES SUR LE MODELE V

Actuellement dans les entreprises, le développement de logiciels peut se calquer sur l'un des modèles les plus connus, principalement les modèles cascade, en V et spirale, et ceci sans suivre de véritable méthode. Dans ce premier cas, l'équipe de développement peut toutefois utiliser dans certaines des étapes des techniques ou outils issus de différentes méthodes, engendrant par là même d'ailleurs certaines confusions. Par exemple, il est fréquent d'entendre des développeurs déclarer utiliser la méthode MERISE alors qu'ils se contentent simplement d'utiliser le modèle entité-association pour décrire les données (!).

<sup>19</sup> Notons à ce sujet qu'un article en commun avec l'équipe de Bernard HOURIEZ portant sur une architecture d'atelier de génie cognitif pour le développement de système à base de connaissance centré sur l'utilisateur a été proposé récemment et accepté dans un congrès sur l'analyse, la conception et l'évaluation de systèmes homme-machine (DURIBREUX, KOLSKI et HOURIEZ, 1995).

L'expérience de l'équipe dans des projets similaires influera fortement sur la qualité du résultat final.

Dans un second cas souvent plus favorable, l'équipe de développement utilise une méthode (MERISE, AXIAL, SASD, SYS\_P\_O...). Même si aucune d'entre elles n'est parfaite, elles possèdent l'avantage d'offrir un cadre de travail rigoureux, de fournir des lignes directrices, de proposer des formalismes, des supports de dialogue, etc.

Dans la mesure où nous proposons un modèle de développement, il nous paraît nécessaire de pouvoir répondre aux besoins d'une équipe de développement dans les deux cas de figure précédents. Ainsi, le modèle doit être suffisamment rigoureux et clair pour se suffire à lui-même. Un premier travail de fond consiste alors à proposer dans chacune des étapes un ensemble de méthodes, de techniques et d'outils opérationnels. Vis-à-vis du second cas de figure, une ou plusieurs méthodes supportant le modèle  $\nabla$  doivent impérativement être proposées.

Un autre travail de fond vise la mise au point de méthodes basées sur le modèle  $\nabla$ . A ce sujet, il n'est pas inutile de rappeler que chaque domaine d'application (bureautique, EAO, CAO, supervision de systèmes complexes...) nécessite la prise en compte de critères, d'aspects, de points de vue spécifiques concernant l'évaluation, la conception et l'utilisation de systèmes interactifs.

C'est pourquoi il ne nous paraît pas réaliste de nous orienter sur une méthode générale et de chercher à l'adapter ensuite aux spécificités du domaine d'application visé. Par exemple, il n'est pas raisonnable d'envisager, comme certains auteurs, l'utilisation de la méthode MERISE (qui donne de bons résultats pour les applications de gestion) pour le développement de systèmes industriels temps réel.

Ainsi, pour chaque domaine d'application, il nous paraît nécessaire de proposer une méthode particulière ayant pour points communs les principes de base du modèle  $\nabla$  ainsi qu'éventuellement certains outils de modélisation du système homme-machine.

Si l'on revient aux systèmes industriels complexes, les travaux menés au laboratoire visant à mettre au point la méthode dite en U sont d'une extrême importance et devraient déboucher à terme sur une méthode opérationnelle allant de la modélisation du système homme-machine à son évaluation systématique. Plusieurs travaux vont actuellement dans ce sens en s'intéressant à des aspects complémentaires dans la méthode en U. Par exemple :

- ABED et ANGUE (1994) modélisent les tâches humaines prescrites et les activités réelles par SADT et les réseaux de Petri, et mettent au point un modèle de confrontation des données recueillies.
- MILLOT et DEBERNARD (à paraître) s'intéressent à la définition des modes de coopération à mettre en place entre les utilisateurs et les modules d'aide à la décision disponibles à partir de l'interface homme-machine.
- GRISLIN, KOLSKI et ANGUE (1993 ; 1995) visent à mettre en place des méthodes d'évaluation a priori à chaque étape du cycle de développement de système interactif.

Lorsque l'ensemble de ces travaux aura conduit à des démarches d'ingénierie immédiatement opérationnelles, il s'agira de les unifier dans le but de proposer une méthode complète assistant l'équipe de développement dans chacune des étapes. En parallèle, il sera intéressant de s'orienter sur d'autres domaines d'application, figure 3.6, et de proposer des méthodes prévues spécifiquement pour les systèmes grand public, l'EAO, la CAO, etc. Une collaboration étroite avec des spécialistes des domaines cités s'avère bien entendu indispensable.

Dans le même ordre d'idée, il me paraîtrait utile de reprendre les principales méthodes existantes (MERISE, AXIAL, OOA/OOD...) et de les remodeler selon l'état d'esprit du modèle  $\nabla$ . Il s'agirait de remédier à leurs lacunes vis-à-vis de des facteurs humains, mais tout en veillant au maximum à garder leurs acquis. Un travail considérable devrait ainsi pouvoir être réalisé.

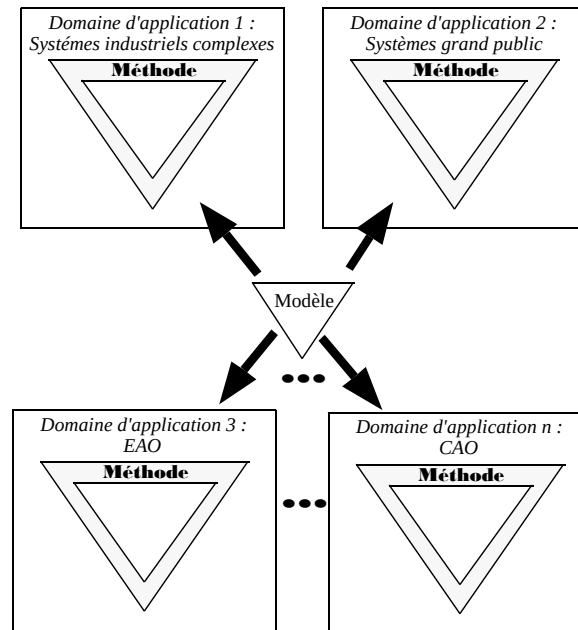


Figure 3.6 : Mise en place de méthodes autour du modèle  $\nabla$

### 3.4. VERS DES OUTILS LOGICIELS ET DES ATELIERS DE GENIE LOGICIEL SUPPORTANT LE MODELE $\nabla$

Pour être véritablement efficace, l'utilisation de chaque méthode de développement basée sur le modèle  $\nabla$  devra être supportée par un atelier de génie logiciel, ou au moins un ensemble d'outils logiciels, offrant des facilités pour chaque étape à suivre.

Dans le processus de développement, plusieurs rôles clés ont été progressivement identifiés dans les deux chapitres précédents. En plus du gestionnaire de projet <sup>20</sup> qu'il convient de citer même si celui-ci n'a pas vraiment fait l'objet de notre travail et a donc été jusqu'à maintenant occulté, on retrouve plusieurs intervenants : le développeur, l'utilisateur, le spécialiste des facteurs humains, l'expert du système et le bibliothécaire. Un intervenant a été rajouté : le cogniticien (ou ingénieur de la connaissance), pièce maîtresse bien connue du développement de systèmes à base de connaissance (GAINES et al., 1993 ; HOURIEZ, 1994), dans la mesure où différents modules d'aide peuvent prendre la forme de systèmes à base de connaissance.

C'est pourquoi on doit retrouver ces différents intervenants autour de l'atelier visé qui doit prendre la forme d'un *atelier coopératif* facilitant le dialogue entre les différents intervenants, figure 3.7.

Cet atelier doit posséder les mêmes fonctionnalités que les ateliers habituels offrant des moyens de spécification, conception et réalisation, ainsi que des moyens de tests et de validation technique (FOUQUE, 1990 ; VESSEY et al., 1992). Mais, en plus, il doit apporter des moyens de modélisation du système homme-machine, par une décomposition en un modèle de référence et un modèle réel, figure 3.1 et apporter des solutions en matière de confrontation socio-ergonomique de ces deux modèles.

<sup>20</sup> Voir à ce sujet le numéro spécial de la revue Communications of the ACM intitulé "Orchestrating project organization and management" (ACM, 1993).

Comme l'indique le modèle, il s'agira aussi pour les étapes de spécification, conception et codage, de fournir des moyens facilitant l'évaluation et la validation des solutions envisagées. Les perspectives de recherche sont énormes à ce sujet. La thèse de Martial GRISLIN prévue pour fin 1994, consacrée à l'utilisation de modèles formels pour l'évaluation prédictive dans le cycle de développement, va dans ce sens.

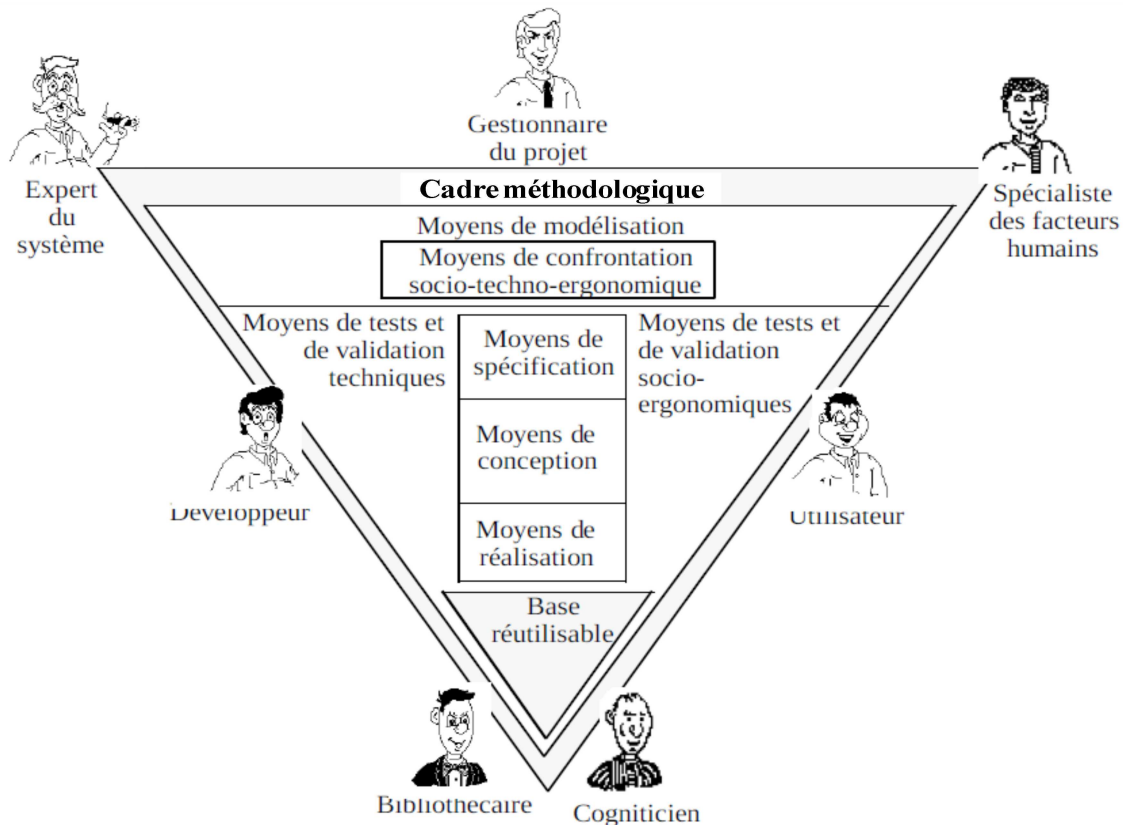


Figure 3.7 : Principe d'un atelier coopératif supportant le modèle  $\nabla$

### 3.5. DU MODELE $\nabla$ AUX INTERFACES DITES "INTELLIGENTES"

Ce mémoire porte sur des systèmes interactifs constitués d'une interface homme-machine à laquelle peuvent être connectés des modules d'aide à la décision. Ces modules d'aide peuvent être éventuellement basés sur des techniques d'intelligence artificielle.

Mais, l'interface homme-machine peut être dite "intelligente" lorsque des modules intégrés au contrôleur du dialogue (au sens du modèle de SEEHEIM) visent à améliorer les interactions homme-machine. Comme on l'a précisé dans les deux chapitres précédents, à part pour les interfaces dites flexibles, les approches d'interfaces "intelligentes" sont généralement basées sur des techniques d'intelligence artificielle également.

Il est donc clair que de la manière dont il est proposé, le modèle  $\nabla$  peut être facilement adapté dans une optique de développement d'interface dite "intelligente". A ce sujet, en accord avec JURAIN et d'autres auteurs, nous avons souligné en discutant d'interfaces qualifiables de "classiques" en introduction de la partie 1.3, que le développement de l'application d'une part et de l'interface d'autre part entraîne la nécessité de développeurs aux compétences complémentaires. Si l'on s'intéresse aux interfaces dites "intelligentes", cette remarque peut encore s'appliquer vis-à-vis de l'interface elle-même. En effet, en s'inspirant du modèle de SEEHEIM, il s'agit encore de redécomposer le développement en étapes parallèles.

Une première proposition d'adaptation du modèle V au développement d'interfaces "intelligentes" est visible en figure 3.8. Sur celle-ci, la partie gauche s'intéresse à la partie visible de l'interface, c'est-à-dire l'aspect présentation. La partie droite du modèle quand à elle prend en charge la partie "transparente" pour l'utilisateur, en l'occurrence le contrôleur de dialogue associé à l'interface avec l'application.

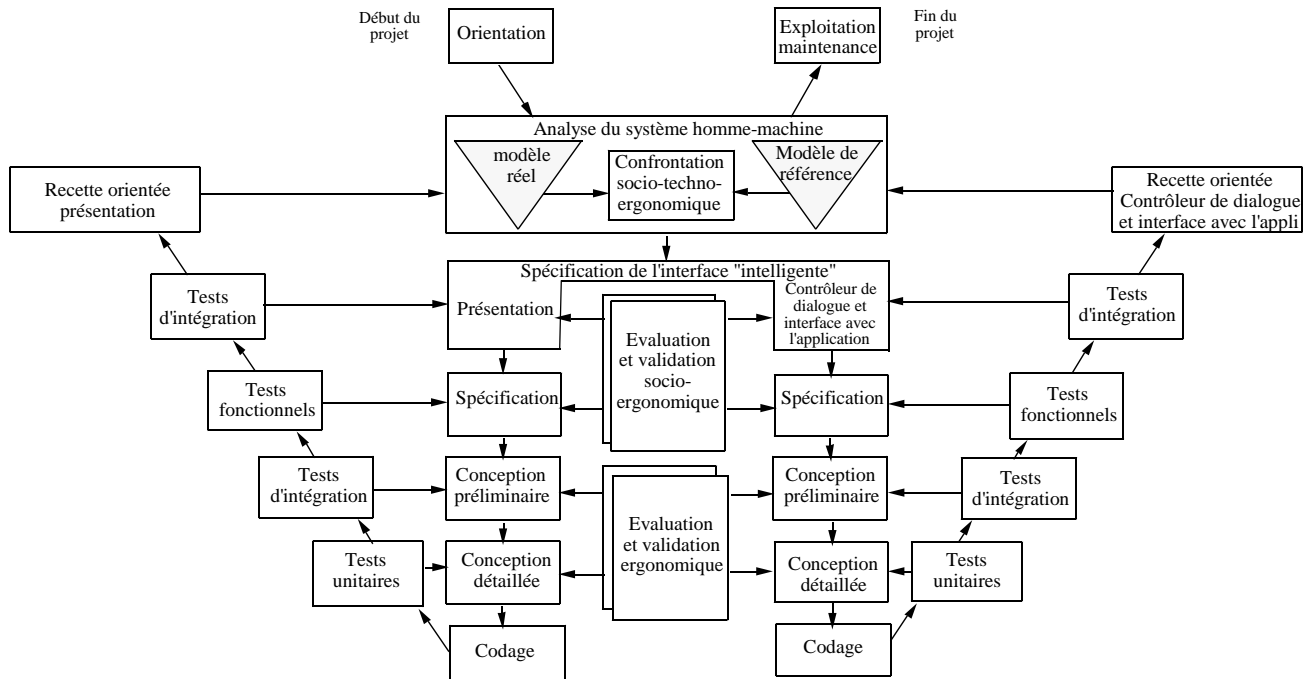


Figure 3.8 : Modèle V de développement d'interface "intelligente"

Bien entendu, de nombreux concepts et architectures d'interfaces "intelligentes" sont possibles pour le versant droit du modèle concernant le contrôleur de dialogue et l'interface avec l'application. Nous souhaitons lors de ces prochaines années de recherche parcourir l'ensemble des architectures possibles, allant de structures de systèmes à base de connaissance classiques à celles issues de l'intelligence artificielle distribuée.<sup>21</sup>

A ce sujet, les systèmes multi-agents offrent des perspectives prometteuses dans le domaine des interfaces dites "intelligentes" et constituent en conséquence un champ de recherche à défricher. Un vaste travail me paraît devoir être mené en collaboration avec des équipes spécialisées en intelligence artificielle distribuée.<sup>22</sup>

Enfin, une interface "intelligente" doit être capable de s'adapter en temps réel aux besoins de l'utilisateur, et ceci sous les contraintes de l'application. Maintenant que de premiers concepts liés aux interfaces "intelligentes" ont été étudiés dans le cadre du projet MDI, il est temps d'aller plus loin en s'intéressant de plus près à la contribution potentielle des nombreux travaux menés sur le raisonnement temporel et temps réel en intelligence artificielle (Mc DERMOTT, 1982 ; ALLEN, 1984 ; SHOHAM, 1987, etc). Il faut noter que beaucoup de maquettes de laboratoire raisonnant sur le temps ont été réalisées, mais peu de systèmes opérationnels ont vu le jour dans l'industrie (Cf. la synthèse de HATON et al., 1991). Dans le but de parvenir à terme à des environnements de développement d'interfaces "intelligentes" exploitables

21 Pour les systèmes multi-agents, le lecteur se référera aux ouvrages édités par HUNS (1987), BOND et GASSER (1988) ou GASSER et HUNS (1989). Notons également deux ouvrages pédagogiques s'intéressant d'abord aux architectures classiques pour s'orienter ensuite sur celles issues de l'IAD : celui de BOY (1988) fortement imprégné du souci d'assistance à l'utilisateur dans les systèmes industriels complexes, et celui de HATON et ses collègues (1991) qui ont développé plusieurs architectures originales de systèmes à base de connaissance. Signalons également les travaux de HUGONNARD et GARBAY (1992) mettant actuellement au point un environnement de génération de systèmes multi-agents pour la résolution de problèmes complexes, appelé MAPS.

22 Rappelons à ce sujet que certaines réflexions menées avec René MANDIAU sur l'apport potentiel des approches multi-agents pour la définition d'architectures d'interfaces "intelligentes" ont été cités précédemment, Cf. 1.3.2.5 et 2.1.4.5.

le plus simplement possible par des ingénieurs, un rapprochement me paraît nécessaire avec des chercheurs pour qui l'aspect temporel joue un rôle important dans le développement de solutions concrètes.

## CONCLUSION

Nos travaux de recherche menés depuis 1985 en ingénierie de conception et d'évaluation des interfaces homme-machine nous ont amené à proposer un modèle de développement de système interactif explicitant conjointement les étapes liées aux aspects techniques et humains à considérer. Ce modèle, appelé  $\nabla$ , n'est pas une méthode finie, opérationnelle, mais doit être considéré comme un schéma directeur pour de futures travaux dans le cadre d'une recherche menée en équipe.

Le modèle  $\nabla$  a permis de positionner un ensemble de concepts tournant autour de la réutilisabilité de composants logiciels pour le développement de l'interface et des modules d'aide disponibles éventuellement à travers celle-ci. Des travaux de recherche sont à mener dans ce domaine en collaboration avec des équipes spécialisées en ingénierie de la connaissance.

Dans un domaine d'application particulier, le modèle  $\nabla$  doit être supporté par une méthode adaptée au domaine visé. Dans les systèmes industriels complexes, les travaux menés au LAMIH concernant la méthode en U devraient déboucher à terme sur une méthode couvrant toutes les étapes du développement d'un système interactif. Pour d'autres domaines, il serait très intéressant d'étudier et améliorer des méthodes existantes. Par exemple, la méthode MERISE pourrait bénéficier d'apports importants concernant les aspects liés aux interfaces homme-machine (Cf. par exemple les travaux menés par BARTHET dans cet état d'esprit depuis 1988).

Une méthode est véritablement efficace si elle peut s'appuyer sur des outils puissants et cohérents. Une perspective de recherche consiste à mettre progressivement en place des outils logiciels supportant le modèle  $\nabla$ , ces outils s'orientant progressivement vers des ateliers de génie logiciel basés sur le modèle.

Les interfaces dites "intelligentes" constituent un de nos pôles d'intérêt. Il s'avère que le modèle  $\nabla$ , moyennant quelques adaptations, fournit un cadre théorique et méthodologique de développement de celles-ci. Dans les années à venir, celui-ci devra être progressivement affiné, et des méthodes spécifiques aux interfaces "intelligentes" devront progressivement être mises au point.<sup>23</sup>

En conclusion, nos perspectives de recherche sont résolument tournées vers l'ingénierie des interfaces homme-machine, le génie logiciel et l'intelligence artificielle. Dans ces trois domaines, plusieurs mots-clés ressortent des perspectives envisagées :

- En ingénierie des interfaces homme-machine, les besoins sont énormes en matière de méthodes de conception et d'évaluation. Un travail conséquent reste encore à accomplir.
- Du génie logiciel se dégagent la modélisation du cycle de développement, la réutilisabilité avec comme principe sous-jacent l'orientation objet, ainsi que les méthodes et ateliers supports à la modélisation  $\nabla$ .
- De l'intelligence artificielle se dégagent les aspects liés à la connaissance et au raisonnement, des architectures classiques à celles se rapprochant des systèmes multi-agents. Ces domaines apportent des approches de solutions pour la mise en place de systèmes à base de connaissance pour l'évaluation et la conception d'interfaces, ainsi que pour la mise en œuvre d'interfaces dites "intelligentes".

<sup>23</sup> En 1992, dans la revue *International Journal of Human Factors in Manufacturing*, nous avons proposé une première approche de méthode de développement d'interface "intelligente" (KOLSKI et al., 1992), qui demande à être peaufinée, retravaillée.

## Bibliographie

### Bibliographie de l'équipe

- DURIBREUX M., KOLSKI C., HOURIEZ B. (1995).** *Towards an integration of cognitive ergonomics concepts in knowledge based system development methodologies.* Proposition acceptée pour le 6th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Man-Machine Systems, M.I.T., Cambridge, USA, June 27-29, 1995.
- GRISLIN M., KOLSKI C., ANGUE J.C. (1993).** *Towards an organization of man-machine interface evaluation techniques by a usability criteria grid.* XII European Annual Conference on Human Decision Making and Manual Control, Kassel, 22-24 June.
- GRISLIN M., KOLSKI C., ANGUE J.C. (en cours de soumission).** *Méthodes et techniques d'évaluation des interfaces homme-machine : revue de synthèse et positionnement dans le cycle de développement de système interactif.* En cours de soumission à la revue Techniques et Sciences Informatiques (T.S.I.), juillet 1994.
- GRISLIN M., KOLSKI C., ANGUE J.C. (1995).** *Human-computer interface evaluation in industrial complex systems : a review of usable techniques.* Proposition acceptée pour le 6th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Man-Machine Systems, M.I.T., Cambridge, USA, June 27-29, 1995.
- GRISLIN M. (à paraître).** *Modèles formels pour le développement d'imagerie industrielle (titre provisoire).* Thèse de doctorat prévue pour fin 1994.
- KOLSKI C., TENDJAOU M., MILLOT P. (1992).** *A process method for the design of "intelligent" man-machine interfaces : case study : "the Decisional Module of Imagery".* International Journal of Human Factors in Manufacturing, vol. 2 (2), 155-175.

### Bibliographie complémentaire

- ABED M., ANGUE J.C. (1994).** *A new method for conception, realisation and evaluation of man-machine.* Proceedings IEEE International Conference on Systems, Man and Cybernetics, San Antonio, Texas, October 2-5.
- ACM (1993).** *Numéro spécial intitulé "Orchestrating Project Organization and Management".* Communications of the ACM, 36 (10), october.
- ALLEN J.F. (1984).** *Towards a general theory of action and time.* Artificial Intelligence, 3 (2), pp. 123-154.
- BARTHET M.F. (1988).** *Logiciels interactifs et ergonomie, modèle et méthodes de conception.* Dunod informatique Paris.
- BOND A.H., GASSER L. (Eds.) (1988).** *Readings in distributed artificial intelligence.* Morgan Kaufman, San Mateo, CA.
- CHANDRASEKARAN B. (1987).** *Towards a functional architecture for intelligence based on generic information processing tasks.* Tenth International Joint Conference on Artificial Intelligence (IJCAI-87), pp. 1183-1192, Milan, august.
- ERMINE J.L. (1993).** *Génie logiciel et génie cognitif pour les systèmes à base de connaissances, volume 1 : aspects méthodologiques, volume 2 : études de cas.* Technique et documentation, Lavoisier.
- FOUQUE G. (1990).** *Les ateliers de génie logiciel, état de l'art.* Rapport de recherche MASI.90.54, Laboratoire de méthodologie et architecture des systèmes informatiques, Paris, décembre.
- GAINES B., SHAW M.L.G. (1993).** *Eliciting knowledge and transferring it effectively to a knowledge-based system.* IEEE Transactions on Knowledge and Data Engineering, 5 (1), pp. 4-14, february.
- GASSER L., HUNS M.N. (Eds.) (1989).** *Distributed artificial intelligence, vol. 2.* Pitman, London.
- HALL P. (ed.) (1992).** *Software Reuse and Reverse Engineering in Practice.* Chapman & Hall, London, 584 pages.
- HATON J.P., BOUZID N., CHARPILLET F., HATON M.C., LAASRI B., LAASRI H., MARQUIS P., MONDOT T., NAPOLI A. (1991).** *Le raisonnement en intelligence artificielle, modèles, techniques et architectures pour les systèmes à bases de connaissances.* InterEditions, Paris.
- HOURIEZ B. (1994).** *Acquisition de connaissances pour l'aide à la conduite et la supervision de procédés industriels.* Rapport d'habilitation à diriger des recherches, université de Valenciennes, février.
- HUGONNARD E., GARBAY C. (1992).** *Elicitation, présentation et résolution : vers une coopération en univers multi-agents.* Revue d'Intelligence Artificielle, 6 (1-2), pp. 131-156.
- HUNS M.N. (Ed.) (1987).** *Distributed artificial intelligence, vol. 1.* Pitman, London.
- IEEE Software (1990).** *Special Issue on Reuse.* IEEE, January.
- Mc DERMOTT D. (1982).** *Temporal logic for reasoning about processes and plans.* Cognitive Science, vol. 6, pp. 101-155.

- MILLOT P., DEBERNARD S. (à paraître).** *Coopération homme-machine : problématique et méthodologie de mise en oeuvre.* Le Travail Humain, numéro spécial intitulé "approches cognitives de la supervision d'environnements dynamiques", publication prévue pour 1994.
- MEIJER B., RAMES E. (1989).** *Software reuse : from theory to practice.* Quatrièmes Journées Internationales "Le Génie Logiciel et ses applications", Toulouse, décembre, Editions EC2.
- MOREL J.M. (1994).** *The REBBOT Approach to Software Reuse.* International Workshop on Software Reuse, Bruxelles, June 8.
- SHOHAM Y. (1987).** *Temporal logics in artificial intelligence : semantical and ontological considerations.* Artificial intelligence, vol. 33, pp. 89-104.
- TRICOT R., KATZ J.Y. (1991).** *Une application industrielle développée en langage à objets.* Quatrièmes Journées Internationales "Le Génie Logiciel et ses applications", Toulouse, décembre, Editions EC2.
- VESSEY I., JARVENPAA S.I., TRACTINSKY N. (1992).** *Evaluation of vendor products : CASE tools as methodology companions.* Communications of the ACM, 35 (4), april.
- VOGEL C. (1988).** *Génie cognitif.* Masson.
- WIELINGA B., SCHREIBER A., BREUKER A. (1992).** *KADS : a modelling approach to knowledge engineering.* Knowledge Acquisition, volume 4, pp. 5-53.
- WIELINGA B., VAN DE VELDE W., SCHREIBER G., AKKERMANS H. (1993).** *Towards an unification of knowledge modelling approaches.* In Second generation expert systems, J.M. David, J.P. Krivine, R. Simmons (Eds.), pp. 299-335, Springer-Verlag.
- ZOROLA-VILLARREAL R., BARTHET M.F., PAVARD B. (1993).** *Une approche d'évaluation par la simulation cognitive des IHM pour le travail coopératif.* Cinquièmes Journées sur l'Ingénierie des Interfaces Homme-Machine, Lyon, 19-20 Octobre 1993.



# Conclusion générale

## *Vers une informatique plus ... "humaine" ?*

Depuis mon arrivée au Laboratoire en septembre 1985, mes travaux de recherche ont porté sur l'ingénierie de conception et d'évaluation des interfaces homme-machine par des approches centrées sur l'utilisateur. Ces approches se sont focalisées sur les systèmes complexes. Ces recherches se situent dans un état d'esprit actuel en informatique, visant à développer des systèmes interactifs en considérant plus finement les caractéristiques et les besoins de l'utilisateur et en l'intégrant à différentes étapes du cycle de vie du logiciel ; voir à ce sujet le bulletin du groupe SIGCHI de l'ACM, et plus particulièrement en France les actes des congrès ERGO-IA (ERGOnomie et Informatique Avancée) ou des Journées sur l'Ingénierie des Interfaces Homme-Machine.

Dans le premier chapitre, l'état de l'art a donné un aperçu de la richesse du domaine, tout en adoptant résolument une démarche pluridisciplinaire. Sans souci d'exhaustivité, de nombreux outils, techniques, modèles et méthodes ont été mis en lumière, issus principalement des sciences de l'ingénieur et des sciences humaines. Ainsi, après un positionnement de la recherche sur le plan méthodologique, l'analyse et la modélisation des tâches humaines et de l'utilisateur ont été abordées. Puis, un ensemble de modèles d'architecture de l'interface homme-machine ont été décrits, ceux-ci s'orientant progressivement vers des concepts d'interfaces qualifiées à tort ou à raison "d'intelligentes". Enfin, la spécification, la réalisation et l'évaluation des interfaces homme-machine ont été passées en revue. Tous ces outils, techniques, modèles et méthodes ont été examinés en laissant une large part à la prise en compte de la dimension humaine lors du développement d'une application interactive.

Le second chapitre consacré à mes activités de recherche et d'encadrement doctoral s'est composé de deux parties :

- La première partie a été consacrée aux systèmes mis en oeuvre au cours de ces dernières années. C'est ainsi qu'on a décrit SYNOP, ERGO-CONCEPTOR, l'outil de prototypage intégré dans l'atelier ATLAS et le Module Décisionnel d'Imagerie (M.D.I.). Chacun de ces systèmes est original par les fonctionnalités qu'il procure, visant de surcroît à considérer certains facteurs liés à l'homme dans leur opérationnalité. Ainsi, des connaissances relatives à la manière de présenter l'information sur écran ont été intégrées dans les bases de connaissance de SYNOP, un système visant l'évaluation automatique "statique" de pages-écrans. Le système ERGO-CONCEPTOR a pour objet de générer "automatiquement" des spécifications d'interface en partant d'un ensemble de données issues d'une analyse du système à superviser. Les fonctionnalités intégrées dans l'atelier ATLAS facilitent le prototypage d'imagerie exploitant une description du système physique. Enfin,

le M.D.I. vise à l'aide de ses bases de connaissance à répondre à trois problèmes ergonomiques : QUOI (quelles informations) présenter, QUAND les présenter, COMMENT les présenter. Même si ces systèmes doivent être considérés avant tout comme des outils de recherche, ils nous ont permis d'aborder la conception et l'évaluation d'interface sous des angles nouveaux. En particulier, ils nous ont incité à envisager un ensemble de liens entre les interfaces homme-machine d'une part, l'intelligence artificielle, le génie logiciel et les sciences cognitives d'autre part.

- La seconde partie a laissé entrevoir la part importante consacrée à la mise en place de méthodes. C'est ainsi qu'autour des systèmes développés, plusieurs méthodes ont été commentées. Le travail réalisé autour du recensement de connaissances expertes a d'abord été expliqué. Il a débouché sur la création de bases de connaissances mais a aussi été abordé dans un but de mise à disposition des développeurs d'interfaces de connaissances livresques et didactiques. Le positionnement de SYNOP et ERGO-CONCEPTOR dans la méthode en U fait également partie de nos apports méthodologiques. Une méthode rapide d'analyse de système existant pour la spécification d'imagerie de même type a ensuite été décrite, de même que la méthode que nous avons mise en place pour créer les bases du M.D.I. en utilisant des techniques issues de l'Apprentissage Symbolique Automatique. Enfin, une méthode de développement d'imagerie axée sur le prototypage, supportée par l'atelier ATLAS, a été expliquée. A l'image des systèmes réalisés, chacune des méthodes décrites se caractérise par l'importance accordée à la prise en compte de la dimension humaine.

C'est autour du *génie logiciel pour les interfaces homme-machine* qu'ont été abordées les perspectives de recherche. Tout en adoptant systématiquement une approche méthodologique (caractéristique du génie logiciel), il s'agira en particulier d'étudier de plus en plus en profondeur l'apport de l'intelligence artificielle et des sciences cognitives pour l'ingénierie de conception et d'évaluation des interfaces. En effet, au vu d'une part de l'état de l'art, et d'autre part des tentatives prometteuses découlant des systèmes que nous avons développés, nous sommes persuadés de l'opérationnalisation possible de nouvelles architectures logicielles considérant de plus en plus explicitement la dimension humaine.

Dans celles-ci devraient apparaître à terme de nouvelles données-clés : tâches humaines à accomplir, objectifs à atteindre, caractéristiques de l'utilisateur à considérer, modes de raisonnement à respecter, modes de coopération homme-machine, machine-machine et homme-homme à mettre en place, etc. En résumé, nous souhaitons contribuer de plus en plus à l'avenir à une informatique présentant un visage plus ... humain dès lors qu'il s'agit de développer des systèmes interactifs où l'homme joue le rôle central.

C'est ainsi que nos perspectives de recherche ont été envisagées autour d'un modèle appelé  $\nabla$  de développement de système interactif. Ce modèle reprend les étapes classiques du génie logiciel que l'on retrouve dans les modèles cascade, spirale ou en V. Toutefois, il explicite la prise en compte de facteurs humains lors du développement des systèmes interactifs, en s'inspirant du cadre théorique et méthodologique défini dans le premier chapitre. Au contraire des modèles existants, le modèle  $\nabla$  vise à situer l'utilisateur dans le développement et à inciter le développeur à aborder le cycle de vie du logiciel avec un état d'esprit différent, et non plus systématiquement orienté sur les facteurs techniques. Plusieurs axes de recherche ont ensuite été précisés autour de la réutilisabilité de composants logiciels, de la mise en place de méthodes, outils et ateliers supportant le modèle  $\nabla$ , et de l'étude d'interfaces dites "intelligentes".

Notons que dans les chapitres de ce mémoire, nous avons choisi d'occulter *pour l'instant* de nouveaux concepts qui ne sont pas encore apparus, ou si peu, dans les interfaces permettant de contrôler un système complexe : par exemple la réalité virtuelle et ses nouveaux modes d'entrée/sortie, l'interaction multimodale consistant à multiplier les moyens d'entrée d'informations dans l'interface et la possibilité de les utiliser simultanément, ou encore le collecticiel permettant la prise en compte du travail en groupe. Mais l'évolution technologique est telle que déjà on voit apparaître de premières applications concrètes de ces nouveaux concepts dans le domaine qui nous préoccupe, citons entre autre l'utilisation d'interfaces multimodales pour le contrôle de la navigation aérienne, ou de gants tactiles en téléopération.

*Pour l'ensemble des axes de recherche envisagés, un travail considérable reste à accomplir, au vu de la place de plus en plus importante que tient l'ingénierie des interfaces homme-machine dans de nombreux domaines...*